

# A Quasi-Linear Time Algorithm Deciding Whether Weak Büchi Automata Reading Vectors of Reals Recognize Saturated Languages

Arthur Milchior

## Abstract

This work considers weak deterministic Büchi automata reading encodings of non-negative  $d$ -vectors of reals in a fixed base. A saturated language is a language which contains all encoding of elements belonging to a set of  $d$ -vectors of reals. A Real Vector Automaton is an automaton which recognizes a saturated language. It is explained how to decide in quasi-linear time whether a minimal weak deterministic Büchi automaton is a Real Vector Automaton. The problem is solved both for the two standard encodings of vectors of numbers: the sequential encoding and the parallel encoding. This algorithm runs in linear time for minimal weak Büchi automata accepting set of reals. Finally, the same problem is also solved for parallel encoding of automata reading vectors of relative reals.

## 1 Introduction

This paper deals with logically defined sets of vector of numbers encoded by Büchi deterministic automata. The sets of vectors of integers whose encodings in base  $b$  are recognized by a finite automaton are called the  $b$ -recognizable sets. By [BHMV94], the  $b$ -recognizable sets are exactly the sets which are  $\text{FO}[\mathbb{Z}; +, <, V_b]$ -definable, where  $V_b(n)$  is the greatest power of  $b$  dividing  $n$ . It was proven in [Sem77, Cob69] that the  $\text{FO}[\mathbb{N}; +]$ -definable sets are exactly the sets which are  $b$ - and  $b'$ -recognizable for every  $b \geq 2$ .

Those results have then been extended to results about sets of vectors of reals recognized by a Büchi automata. The notion of Büchi automata is a formalism which describes languages of infinite words, also called  $\omega$ -words. The Büchi automata are similar to the finite automata. The main difference between the two kinds of automata is that a finite automaton accepts a finite word if it admits a run ending on accepting states, while a Büchi automaton accepts an infinite word if it admits a run in which an accepting state appears infinitely often.

One of the main differences between finite and Büchi automata is that finite automata can be determinized while deterministic Büchi automata are less expressive than Büchi automata. For example, the language  $L_{\text{fin } a}$  of words containing a finite number of times the letter  $a$  is recognized by a Büchi automaton, but is not recognized by any deterministic Büchi automaton. This statement implies, for example, that no deterministic Büchi automaton recognizes the set of reals of the form  $nb^p$  with  $n \in \mathbb{N}$  and  $p \in \mathbb{Z}$ , that is, the reals which admits no encoding in base  $b$  with a finite number of non-0 digits.

Another main difference between the two classes of automata is that the class of languages recognized by finite automata is closed under complement while the class of languages recognized by deterministic Büchi automata is not closed under complement. For example,  $L_{\text{inf } a}$ , the complement of  $L_{\text{fin } a}$ , is recognized by a deterministic Büchi automaton.

Real numbers are naturally encoded, in a base  $b > 1$ , as a sequence of digits in  $\{0, \dots, b-1\}$  and a separator symbol  $\star$ . That is, as a word over the alphabet  $\{0, \dots, b-1, \star\}$ . Similarly, a  $d$ -vector of real numbers can be encoded as a word over alphabet  $\{0, \dots, b-1\}^d \cup \{\star\}$ , where  $d$  digits are read simultaneously, one for each element of the vector. This is call the  $d$ -parallel encoding of the vector. A  $d$ -vector can also be encoded as a word over alphabet  $\{0, \dots, b-1, \star\}$ , assuming that the digits in position  $i$  modulo  $d$  corresponds to the digits of the  $i$ -th element of the vector. This is call the sequential encoding of the vector of digits. The cardinality of the alphabet of parallel encoding is exponentially bigger than the cardinality of the alphabet of sequential encodings, thus, sequential encodings may be preferred for practical purposes. Parallel encoding leads to simpler notation, hence, most of the litterature consider parallel encodings. We consider both encodings in this paper.

A language  $L$  is said to be *saturated* if, given a vector  $\mathbf{r} \in \mathbb{R}^d$ , the set of its encoding in base  $b$  is either included in  $L$  or disjoint from  $L$ . A Real Vector Automaton (RVA, See e.g. [BBL09]) is an automaton of alphabet  $\{0, \dots, b-1\}^d \cup \{\star\}$  which recognizes a saturated language. Here  $d$  is the dimension of the vector that the automata read. In the case where the dimension  $d$  is 1, those automata are called Real Number Automata (RNA, See e.g. [BBB10]).

The sets of vectors of reals whose encodings in base  $b$  is recognized by a RVA are called the  $b$ -recognizable sets. By [WB00], they are exactly the  $\text{FO}[\mathbb{R}, \mathbb{Z}; +, <, X_b, 1]$ -definable sets. The logic  $\text{FO}[\mathbb{R}, \mathbb{Z}; +, <, X_b, 1]$  is the first-order logic over reals with a unary predicate which holds over integers, addition, order, the constant one, and the function  $X_b(x, u, k)$ . The function  $X_b(x, u, k)$  holds if and only if  $u$  is equal to some  $b^n$  with  $n \in \mathbb{Z}$  and there exists a encoding in base  $b$  of  $x$  whose digit in position  $n$  is  $k$ . That is,  $u$  and  $x$  are of the form:

$$\begin{array}{lcl} u = & 0 & \dots & 0 & \star & 0 & \dots & 0 & 1 & 0 & \dots & \text{or} & u = & 0 & \dots & 0 & 1 & 0 & \dots & 0 & \star & 0 & \dots \\ x = & & & & \star & & & & k & & & & x = & & & & k & & & & \star & & \dots \end{array}$$

A weak deterministic Büchi automaton is a deterministic Büchi automaton whose set of accepting states is a union of strongly connected components. A set is said to be weakly  $b$ -recognizable if it is recognized by a weak automaton in base  $b$ . By [BBL09], a set is  $\text{FO}[\mathbb{R}, \mathbb{Z}; +, <]$ -definable if and only if its set of encodings is weakly  $b$ -recognizable for all  $b \geq 2$ . The weak deterministic Büchi automata are less expressive than the deterministic Büchi automata. For example, the language  $L_{\inf a}$  of words containing an infinite number of  $a$  is recognized by a deterministic Büchi automaton but is not recognized by any weak deterministic Büchi automaton. This implies that, for example, no weak deterministic Büchi automaton recognizes the set of reals which are not of the form  $nb^p$  with  $n \in \mathbb{N}$  and  $p \in \mathbb{Z}$ , since those reals are the ones whose encodings in base  $b$  contains an infinite number of non-0 digits. Furthermore, by [L01], weak deterministic Büchi automata can be efficiently minimized.

In this paper, we show that we can efficiently decide whether a weak Büchi automaton accept a saturated set of vectors of integers. Furthermore, we give an algorithm for automata reading parallel encoding and for automata reading sequential encoding.

We recall standard definition in Section 2. We introduce encoding of sets of vectors of numbers in Section 3. We introduce Büchi automata in Section 4. We formalize how we compute the complexity of an algorithm in Section 5. We study automata reading vectors of numbers in Section 6. We study how to transform words and automata in Section 7. We characterize the parallel RVA in Section 8 and the sequential RVA in Section 9. We explain how to decide whether an automaton is a RVA in Section 10. The case of sets containing negative reals is discussed in Section 11.

## 2 Standard definitions

We now give the standard definitions used in this paper.

**Numbers.** Let  $\mathbb{N}$  and  $\mathbb{R}$  denote the set of non-negative integers and the set of reals, respectively. For  $R \subseteq \mathbb{R}$ , let  $R^{\geq 0}$  denote the set of non-negative elements of  $R$ . For  $n \in \mathbb{N}$ , let  $[n]$  represent  $\{0, \dots, n\}$ . For  $m \in \mathbb{N}^{>0}$ , let  $(n \bmod m)$  represents the only integer  $i \in [m-1]$  such that  $n \equiv i \bmod m$ .

**Sets.** For  $S$  and  $T$  two sets, let  $S \otimes T = \{(s, t) \mid s \in S, t \in T\}$  be the set of ordered pair containing an element of  $S$  and an element of  $T$ . Let  $|S|$  be the cardinality of  $S$ . For  $d \in \mathbb{N}$ , let  $S^d$  be the set of  $d$ -vectors of elements of  $S$  for  $d \in \mathbb{N}$ . The  $d$ -vectors are denoted  $\mathbf{s} = (s_0, \dots, s_{d-1})$  with each  $s_i \in S$ . The  $d$ -vector  $(0, \dots, 0)$  is denoted  $\mathbf{0}$ .

**Finite and infinite words.** An *alphabet* is a finite set, its elements are called *letters*. A finite word over the alphabet  $A$  is a finite sequence of letters of  $A$ . An  $\omega$ -word over the alphabet  $A$  is an infinite sequence of letters of  $A$ . The empty word is denoted  $\epsilon$ . A set of finite (respectively  $\omega$ -) words of alphabet  $A$  is called a language (respectively, an  $\omega$ -language) over alphabet  $A$ .

Let  $w$  be a word, its length is denoted  $|w|$ , it is either a non-negative integer or the cardinality of  $\mathbb{N}$ . For  $n \in [|w| - 1]$ , let  $w[n]$  denote the  $n$ -th letter of  $w$ . For  $v$  a finite word, let  $u = vw$  be the *concatenation* of  $v$  and of  $w$ , that is, the word of length  $|v| + |w|$  such that  $u[i] = v[i]$  for  $i \in [|v| - 1]$  and  $u[|v| + i] = w[i]$  for  $i \in [|w| - 1]$ . Let  $w[< n]$  denote the *prefix* of  $w$  of length  $n$ , that is, the word  $u$  of length  $n$  such that  $w[i] = u[i]$  for all  $i \in [n - 1]$ . Similarly, let  $w[\geq n]$  denote the *suffix* of  $w$  without its  $n$ -th first letters, that is, the word  $u$  such that  $u[i] = w[i + n]$  for all  $i \in [|w| - n]$ . Note that  $w = w[< i]w[\geq i]$  for all  $i \in [|w| - 1]$ .

**Languages** A language is a set of words. Let  $L$  be a language of finite words and let  $L'$  be either an  $\omega$ -languages or a language of finite words. Let  $LL'$  be the set of concatenations of the words of  $L$  and of  $L'$ . For  $i \in \mathbb{N}$ , let  $L^i$  be the concatenations of  $i$  words of  $L$ . Let  $L^* = \bigcup_{i \in \mathbb{N}} L^i$ , more generally, for  $d, j \in \mathbb{N}$ , let  $L^{d\mathbb{N}+j} = \bigcup_{i \in \mathbb{N}} L^{di+j}$  and  $L^+ = \bigcup_{i > 0} L^i$ . If  $L$  is a set of non-empty word, let  $L^\omega$  be the set of infinite sequences of elements of  $L$ . Finally, let  $L^\infty = L^* \cup L^\omega$ .

## 3 Encoding of set of vectors of numbers

In this section we explain how to encode sets of vectors of numbers using languages. We consider natural and real numbers in Section 3.1. We consider the special case of rationals in Section 3.2. We then consider vectors of reals in Section 3.3. Finally, we consider sets of vectors of reals in Section 3.4.

### 3.1 Encoding of numbers

Let us now consider the encoding of numbers in an integer base  $b \geq 2$ . Let  $\Sigma_b$  be equal to  $[b-1]$ , it is the set of digits. The base  $b > 1$  is fixed for the remaining of this paper. Formally, for  $v \in \Sigma_b^*$  and  $w \in \Sigma_b^\omega$ :

$$[v]_b^I = \sum_{i=0}^{|v|-1} b^{|v|-1-i} v[i] \text{ and } [w]_b^F = \sum_{i=0}^{\infty} b^{-i-1} w[i].$$

Let  $w$  be an  $\omega$ -word with exactly one  $\star$ . It is of the form  $w = w_I \star w_F$ , with  $w_I \in \Sigma_b^*$  and  $w_F \in \Sigma_b^\omega$ . The word  $w_I$  is called the natural part of  $w$  and the  $\omega$ -word  $w_F$  is called its fractional part. We then define:

$$[w_I \star w_F]_b^{\mathbb{R}} = [w_I]_b^I + [w_F]_b^F.$$

Examples of representation of numbers are now given.

$$\begin{array}{cccccc} [(10)^\omega]_2^F = \frac{2}{3} & [(01)^\omega]_2^F = \frac{1}{3} & [0(10)^\omega]_2^F = \frac{1}{3} & [0(1)^\omega]_2^F = \frac{1}{2} & [1(0)^\omega]_2^F = \frac{1}{2} \\ [10]_2^I = 2 & [1]_2^I = 1 & [01]_2^I = 1 & [\epsilon]_2^I = 0 & [00000]_2^I = 0 \\ [10 \star (10)^\omega]_2^{\mathbb{R}} = \frac{8}{3} & [\star 0(1)^\omega]_2^{\mathbb{R}} = \frac{1}{2} & & [00000 \star 1(0)^\omega]_2^{\mathbb{R}} = \frac{1}{2}. \end{array}$$

**Pair-encoding** A word  $w \in \Sigma_b \cup \{\star\}^\infty$  can equivalently be encoded as a pair  $\langle w, S \rangle$  where  $w \in \Sigma_b^\infty$  and  $S \subseteq \mathbb{N}$ . This pair represents the word of length  $|w| + |S|$ , such that  $\langle w, S \rangle[i]$  is  $\star$  if  $i \in S$ , otherwise it is  $w[i - |\{s \in S \mid s < i\}|]$ . Intuitively, for  $i \notin S$ , the  $i$ -th letter of  $\langle w, S \rangle[i]$  is the letter of  $w$  at a position  $j$  such that  $j + k = i$ , where  $k$  is the number of  $\star$ 's before  $i$ . For example  $\langle (10)^\omega, \{2\} \rangle = 10 \star (10)^\omega$ ,  $\langle 01, \{0\} \rangle = \star 01$  and  $\langle (01)^\omega, \emptyset \rangle = (01)^\omega$ . The representation  $\langle w, S \rangle$  is called a pair-encoding.

If  $\langle w, S \rangle$  is an encoding of a real, or an encoding of a factor of a real,  $S$ 's cardinality is at most 1. In particular, the pair encoding of a word of  $\Sigma_b^* \star \Sigma_b^\omega$  is of the form  $\langle w, \{s\} \rangle$  with  $w \in \Sigma_b^\omega$  and  $s \in \mathbb{N}$ . Note however that in order to check whether an automaton is a RVA, it must be checked that it rejects every words whose number of  $\star$ 's is not 1. Therefore, the cases where  $S$  is not a singleton must be considered.

### 3.2 Encoding of rationals

We now recall a basic fact about encoding of rationals.

**Theorem 3.1** ([HW60]). *Let  $q \geq 0$  a real. Let  $l = \lceil \log_b(q + 1) \rceil$  and  $l' \in \mathbb{N}$ .*

*The number of encoding of  $q$  with a natural part of length  $l'$  is:*

- 0 if  $l' < l$ ,
- 2 if  $l' \geq l$  and if  $q$  admits a decomposition of the form  $nb^p$  with  $p \in \mathbb{Z}$  and  $n \in \mathbb{N}$  such that  $n \not\equiv 0 \pmod{b}$ , and
- 1 otherwise.

*In the seconde case, the two encodings are of the form:*

$$(va(b-1)^\omega, \{i\}) \text{ and } (v(a+1)0^\omega, \{i\}), \tag{1}$$

*with  $v \in \Sigma_b^*$  and  $a \in \Sigma_b \setminus \{b-1\}$ .*

This theorem illustrates that pair-encoding leads to shorter statements. Indeed, with the standard-encoding, Equation (1) would require to consider three cases, depending on whether  $p < 0$ ,  $p = 0$  or  $p > 0$ . Note that the condition  $n \not\equiv 0 \pmod{b}$  ensures that  $q \neq 0$ .

### 3.3 Encoding of vectors of reals.

It is now explained how to encode  $d$ -vectors of real numbers. In this paper, we fix a positive integer constant  $d$ . In the remaining of this paper, we only consider sets of dimension 1 or  $d$ .

There exists two standard encodings of vectors of numbers. The parallel one and the sequential one. A parallel encoding consists in a sequence of  $d$ -vector of digits. A sequential encoding

consists in a sequence of digits. This sequence contains alternatively a digit of the zeroth number, a digit of the first number, up to a digit of the  $(d-1)$ -th number. In both cases, exactly one dot appear in the sequence, to separate the natural part from the fractional part.

The alphabet of sequential encoding contains  $(b+1)$  letters while the alphabet of parallel encodings contains  $(b^d)+1$  letters. Thus, sequential encoding allow to create smaller automata, as shown in Example 6.2. However, parallel encoding leads to notations which are more compact. Since parallel encoding are more standards and lead to simpler proofs.

**Parallel encodings** We now introduce the notion of parallel encoding of a  $d$ -vector of numbers. Let  $\Sigma_{b,d} = \Sigma_b^d$ , be the set of  $d$ -vectors of digits. For  $\mathbf{w} \in \Sigma_{b,d}^\infty$ , and  $0 \leq i < d$ ,  $w_i$  denote the unique word such that  $|w_i| = |\mathbf{w}|$  and such that for  $0 \leq k < |\mathbf{w}|$ ,  $(w_i)[k] = (\mathbf{w}[k])_i$ . Similarly, for  $\langle \mathbf{w}, S \rangle \in (\Sigma_{b,d} \cup \{\star\})^\infty$ ,  $\langle \mathbf{w}, S \rangle_i$  denote  $\langle w_i, S \rangle$ .

For  $\mathbf{v} \in \Sigma_{b,d}^*$  and  $\mathbf{w} \in \Sigma_{b,d}^\omega$ , we define  $[\mathbf{v}]_b^I$  as  $([v_0]_b^I, \dots, [v_{d-1}]_b^I)$  and  $[\mathbf{w}]_b^F$  as  $([w_0]_b^F, \dots, [w_{d-1}]_b^F)$ . Similarly, we define  $[\mathbf{v} \star \mathbf{w}]_b^\mathbb{R}$  as  $([v_0 \star w_0]_b^\mathbb{R}, \dots, [v_{d-1} \star w_{d-1}]_b^\mathbb{R}) = [\mathbf{v}]_b^I + [\mathbf{w}]_b^F$  where addition is defined component by component. For example:

$$\left[ \left( \begin{pmatrix} 1 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix}^\omega, \{2\} \right) \right]_b^\mathbb{R} = \left[ \begin{pmatrix} 1 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \end{pmatrix} \right]_b^I + \left[ \begin{pmatrix} 1 \\ 0 \end{pmatrix}^\omega \right]_b^F = (2, 0) + \left( \frac{2}{3}, \frac{2}{3} \right) = \left( \frac{8}{3}, \frac{2}{3} \right). \quad (2)$$

**Sequential encodings** We now introduce the notion of sequential encodings of a  $d$ -vector of numbers. Let  $w \in \Sigma_b^\infty$  whose length is either a multiple of  $d$  or infinite. Let  $\text{par}_d(w) \in \Sigma_{b,d}$  be the only word of length  $|w|/d$ , whose  $i$ -th letter is  $(w_{di+0}, \dots, w_{di+(d-1)})$  for  $i < |w|/d$ . For  $\langle w, S \rangle$ , with  $S$  a set of multiple of  $d$ , let  $\text{par}_d(\langle w, S \rangle) = \langle \text{par}_d(w), \{s/d \mid s \in S\} \rangle$ . Let  $\text{seq}_d(\mathbf{w})$  be the inverse of the function  $\text{par}_d(w)$ . Given a word  $w$ ,  $\text{par}_d(w)$  is called the *parallelization* of  $w$  and  $\text{seq}_d(w)$  is called its *sequentialization*. For example, the parallelization of  $(100(01)^\omega, \{4\})$  is  $\left( \begin{pmatrix} 1 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix}^\omega, \{2\} \right)$ . As seen in Equation (2), it encodes the pair of reals  $(8/3, 2/3)$ .

### 3.4 Encoding of sets of vectors of reals

We now explain how to encode sets of tuples of reals as a language.

**$d$ -parallel languages** The subsets of  $\Sigma_{b,d}^* \star \Sigma_{b,d}^\omega$  are called  *$d$ -parallel language*. Given a  $d$ -parallel language  $L$ , let  $[L]_b^\mathbb{R}$  be the set of vectors of reals admitting an encoding in  $L$ . Formally,  $[L]_b^\mathbb{R} = \{[\mathbf{w}]_b^\mathbb{R} \mid \mathbf{w} \in L\}$ . The language  $L$  is said to be a  *$d$ -parallel encoding* of the set of reals  $[L]_b^\mathbb{R}$ . A  $d$ -parallel language  $L \subseteq \Sigma_{b,d}^* \star \Sigma_{b,d}^\omega$  is said to be *saturated* if, for any  $d$ -vector of numbers  $\mathbf{r} \in [L]_b^\mathbb{R}$ , all encodings in base  $b$  of  $\mathbf{r}$  belongs to  $L$ .

In general, a set of reals may have infinitely many encodings in base  $b$ . For example, for  $I \subseteq \mathbb{N}$  an arbitrary set, the language  $\{0, 1\}^* \star (\{0, 1\}^\omega \setminus \{0^i 1^\omega \mid i \in I\})$  is an encoding in base 2 of  $\mathbb{R}^{\geq 0}$ . It is saturated only for  $I = \emptyset$ .

**$d$ -sequential languages** The case of  $d$ -sequential encodings of vectors of reals is now considered. The subsets of  $\Sigma_b^{d\mathbb{N}} \star \Sigma_b^\omega$  are called  *$d$ -sequential languages*. The parallelization of a  $d$ -sequential language  $L$ , denoted  $\text{par}_d(L)$  is  $\{\text{par}_d(w) \mid w \in L\}$ . A  $d$ -sequential language  $L$  is said to be a  *$d$ -sequential encoding* of the set  $[\text{par}_d(L)]_b^\mathbb{R}$ . This language is said to be saturated if  $\text{par}_d(L)$  is saturated.

## 4 Deterministic Büchi automata

This paper deals with deterministic Büchi automata. We define this notion in Section 4.1. We consider the notion of quotient and morphism of Büchi automata in Section 4.2.

### 4.1 Definition

A *deterministic Büchi automaton* is a 5-tuple  $(Q, A, \delta, q_0, F)$ , with  $Q$  a finite set of states,  $A$  an alphabet,  $\delta : Q \otimes A \rightarrow Q$  is the *transition function*,  $q_0 \in Q$  is the *initial state* and  $F \subseteq Q$  is the set of *accepting states*. For each  $q \in Q$  and  $a \in A$ ,  $q$  is said to be a *predecessor* of  $\delta(q, a)$ . For  $q \in Q$ , let  $\mathcal{A}_q$  be the automaton  $(Q, A, \delta, q, F)$ , that is  $\mathcal{A}$  with  $q$  as initial state. A state  $q \in Q$  is said to be *accessible* from a state  $q' \in Q$  if there exists a finite non-empty word  $w \in A^+$  such that  $\delta(q', w) = q$ . The *strongly connected component* of a state  $q$  is the set of states  $q'$  such that  $q'$  is accessible from  $q$  and  $q$  is accessible from  $q'$ .

From now on in this paper, all Büchi automata are assumed to be deterministic. The function  $\delta$  is implicitly extended on  $Q \otimes A^*$  by  $\delta(q, \epsilon) = q$  and  $\delta(q, aw) = \delta(\delta(q, a), w)$  for  $a \in A$  and  $w \in A^*$ . An example of Büchi automaton is now given.

**Example 4.1.** Let  $\mathcal{A}$  be the automaton pictured in Figure 1. Its alphabet is  $\Sigma_3 \cup \{\star\}$ .

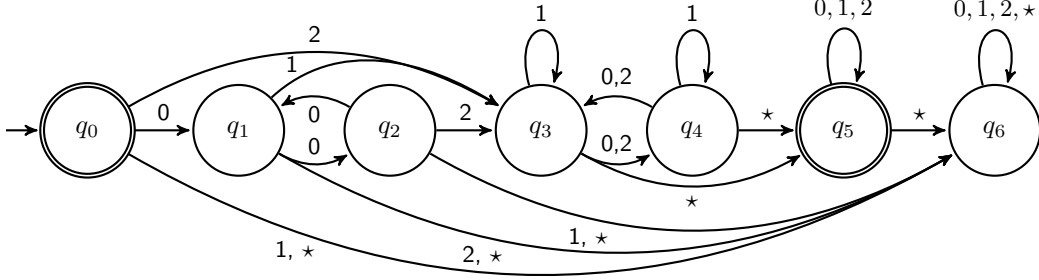


Figure 1: An automaton which recognizes  $(00)(2 + 01)\Sigma_3^* \star \Sigma_3^\omega$ .

Let  $\mathcal{A}$  be an automaton and  $w$  be an infinite word. A *run*  $\pi$  of  $\mathcal{A}$  on  $w$  is a mapping  $\pi : \mathbb{N} \mapsto Q$  such that  $\pi(0) = q_0$  and  $\delta(\pi(i), w[i]) = \pi(i + 1)$  for all  $i < |w|$ . The run is accepting if there exists a state  $q \in F$  such that there is an infinite number of  $i \in \mathbb{N}$  such that  $\pi(i) = q$ . Example 4.1 is now resumed. Note that, if  $\mathcal{A}$  is an automaton, for all  $w \in A^*$  and  $w' \in A^\omega$ , the word  $w'$  is accepted by  $\mathcal{A}_{\delta(q_0, w)}$  if and only if  $ww'$  is accepted by  $\mathcal{A}$ . It is said that  $\mathcal{A}$  recognize the language of words  $w$  such that  $\mathcal{A}$  accepts  $w$ . This language is denoted  $L_\omega(\mathcal{A})$ .

**Example 4.2.** Let  $\mathcal{A}$  be the automaton pictured in Figure 1. The run of  $\mathcal{A}$  on  $01^\omega$  is  $(q_0, q_1, q_3, \dots)$ , with the last state repeated infinitely often. The Büchi automaton  $\mathcal{A}$  does not accept  $01^\omega$  since this run contains exactly one accepting state.

The run of  $\mathcal{A}$  on  $2 \star 1^\omega$  is  $(q_0, q_3, q_5, \dots)$  with the last state repeated infinitely often. The Büchi automaton  $\mathcal{A}$  accepts  $2 \star 1^\omega$  since the accepting state  $q_5$  appears infinitely often in the run.

This automaton recognizes the language  $(00)^*(01, 2)\Sigma_3^* \star \Sigma_3^\omega$ .

### 4.2 Quotients, Morphisms and Weak Büchi Automata

Let  $\mathcal{A} = (Q, A, \delta, q_0, F)$  and  $\mathcal{A}' = (Q', A, \delta', q'_0, F')$  be two Büchi automata. The Büchi automaton  $\mathcal{A}$  is said to be *minimal* if, for each distinct states  $q$  and  $q'$  of  $\mathcal{A}$ ,  $L_\omega(\mathcal{A}_q) \neq L_\omega(\mathcal{A}_{q'})$ . If  $\mathcal{A}'$  is

minimal, a surjective function  $\mu : Q \rightarrow Q'$  is a *morphism* of Büchi automata if  $\mu(q_0) = q'_0$  and if, for each  $q \in Q$ ,  $L_\omega(\mathcal{A}_q) = L_\omega(\text{Aut}'_{\mu(q)})$ . Note that if  $\mu$  is a morphism of Büchi automaton from  $\mathcal{A}$  to  $\mathcal{A}'$ , if  $q$  is a state of  $\mathcal{A}$ , then  $\mu$  is a morphism of Büchi automaton from  $\mathcal{A}_q$  to  $\text{Aut}'_{\mu(q)}$ .

The Büchi automaton  $\mathcal{A}$  is said to be *weak* if  $F$  is a union of strongly connected components. The main theorem concerning quotient of weak Büchi automata is now recalled.

**Theorem 4.3** ([Lö01]). *Let  $\mathcal{A}$  be a weak Büchi automaton with  $n$  states such that all states of  $\mathcal{A}$  are accessible from its initial state. Let  $c$  be the cardinality of  $A$ . There exists a minimal weak Büchi automaton  $\mathcal{A}'$  such that there exists a morphism of automaton  $\mu$  from  $\mathcal{A}$  to  $\mathcal{A}'$ . The automaton  $\mathcal{A}'$  and the morphism  $\mu$  are computable in time  $O(n \log(n)c)$  and space  $O(nc)$ .*

Example 4.1 is now resumed.

**Example 4.4.** Let  $\mathcal{A}_R$  be the Büchi automaton pictured in Figure 1. The automaton  $\mathcal{A}_R$  is weak. Its minimal quotient is pictured in Figure 2. Note that this quotient is not a quotient of finite automata since the accepting state  $q_0$  is sent to a non-accepting state.

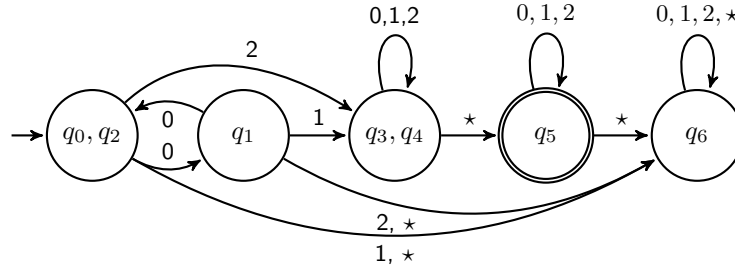


Figure 2: The minimal automaton which recognizes  $(00)(2 + 01)\Sigma_3^* \Sigma_3^\omega$ .

We now explain how to decide efficiently whether two states of two automata recognize the same language.

**Corollary 4.5.** *Let  $A$  an alphabet with  $c > 1$  letters. Let  $\mathcal{A}^0 = (Q^0, A, \delta^0, q_0^0, F^0)$ ,  $\mathcal{A}^1 = (Q^1, A, \delta^1, q_0^1, F^1)$  be weak Büchi automata. Let  $n = |Q^0| + |Q^1|$ .*

*We can compute in time  $O(n \log(n)c)$  and space  $O(nc)$  a data-structure of size  $O(nc)$  such that, for each pair  $(q^0, q^1) \in Q^0 \otimes Q^1$ , we can check in constant time and space whether  $\mathcal{A}_{q^0}^0$  and  $\mathcal{A}_{q^1}^1$  accepts the same language.*

*Proof.* Up to changing the name of the states, we can assume that  $Q^0$  and  $Q^1$  are disjoint. Let  $\alpha \in A$ . Let  $\mathcal{A}' = (Q^0 \cup Q^1 \cup \{q_0\}, A, \delta', q_0^0, F^0 \cup F^1)$ , where  $\delta'(q_0, \alpha) = q_0^0$ ,  $\delta'(q_0, a) = q_0^1$  for each  $a \in A \setminus \{\alpha\}$ , and  $\delta'(q, a) = \delta^i(q, a)$  for  $i \in \{0, 1\}$ ,  $a \in A$  and  $q \in Q^i$ . Clearly  $\mathcal{A}_q^i$  accepts the same language than  $\mathcal{A}'_q$ , for each  $q \in Q^i$ . The automaton  $\mathcal{A}'$  is clearly weak, thus it admits a minimal quotient and a morphism  $\mu$  to this minimal quotient. By Theorem 4.3, this morphism is computable in time  $O(n \log(n)c)$  and takes space  $O(nc)$ . This morphism is the data structure mentioned above.

Let  $(q^0, q^1) \in Q^0 \otimes Q^1$ . Remark that  $\mathcal{A}_{q^0}^0$  and  $\mathcal{A}_{q^1}^1$  accepts the same language if and only if  $\mu(q^0) = \mu(q^1)$ . Given  $\mu$ , this equality can be checked in constant time and space.  $\square$

In practice, the algorithm of [Lö01] could be directly applied to multiple Büchi automata simultaneously. Indeed, the initial state is not considered differently than any other state in this algorithm. Furthermore, this algorithm does not require all states of the automata to be accessible from the initial state.

## 5 Time and space analysis

We now state our assumption above the time and space complexity of in this paper. We first consider the size of the object we use.

All integers and Booleans takes constant space. The size of an automata is the product of the cardinalities of its alphabet and of its set of states. An array of  $n$  elements takes size  $n$ , plus the size of its elements. For a set  $S$  of cardinality  $n$ , a subset of  $S$  is an array of  $n$  Boolean values.

For the sake of simplicity, it is assumed that all basic arithmetic operations over integers, such as addition, multiplication, subtraction, comparison of integers, can be computed in constant time and space. The transition functions of automata return in constant time and space. Creating an array and editing one of its position takes constant time.

## 6 Automata reading set of vectors of reals

We consider automata reading set of vectors of reals in this section. Those automata are formally introduced in Section 6.1. We explain in Section 6.2 how to decide whether an automaton accept a  $d$ -parallel or a  $d$ -sequential language.

### 6.1 Definition

The notion of Büchi automata recognizing a set of vector of reals is now introduced.

A Büchi automaton accepting a  $d$ -parallel or a  $d$ -sequential language is said to be a  $d$ -parallel or a  $d$ -sequential automaton respectively. The set of weak  $d$ -parallel and of weak  $d$ -sequential Büchi automata are closed under taking quotient. The set of  $d$ -vectors of automata associated to an automaton is now introduced.

**Notation 6.1** ( $[\mathcal{A}]_b^{\mathbb{R}}$ ). For  $\mathcal{A}$  a  $d$ -parallel or a  $d$ -sequential automaton, let  $[\mathcal{A}]_b^{\mathbb{R}}$  be  $[L_\omega(\mathcal{A})]_b^{\mathbb{R}}$ .

The following example show that the minimal  $d$ -sequential automaton accepting a set  $R \subseteq (\mathbb{R}^{\geq 0})^d$  can be exponentially smaller than the minimal  $d$ -parallel automaton accepting it.

**Example 6.2.** The minimal  $d$ -parallel automaton accepting  $(\mathbb{R}^{\geq 0})^d$  is:

$$\mathcal{A}^{\text{par}} = (\{q_{\infty, \mathcal{A}}, q_{[0,1], \mathcal{A}}, q_{\emptyset, \mathcal{A}}\}, \Sigma_{2,d} \cup \{\star\}, \delta, q_{\infty, \mathcal{A}}, \{q_{[0,1], \mathcal{A}}\}),$$

where  $\delta(q, \mathbf{a}) = q$  for each state  $q$ , and each letter  $\mathbf{a} \in \Sigma_{2,d}$  and where  $\delta(q_{\infty, \mathcal{A}}, \star) = q_{[0,1], \mathcal{A}}$ . If  $\delta(q, \mathbf{a})$  is not defined above, it is equal to  $q_{\emptyset, \mathcal{A}}$ . This Büchi automaton has 3 states and its alphabet has  $2^d + 1$  letters, hence its size is  $O(2^d)$ . The automaton  $\mathcal{A}^{\text{par}}$  is pictured in Figure 3a, without its state  $q_{\emptyset, \mathcal{A}}$ .

The minimal  $d$ -sequential Büchi automaton accepting  $(\mathbb{R}^{\geq 0})^d$  is:

$$\mathcal{A}^{\text{seq}} = (\{q_{\emptyset, \mathcal{A}}, q_{[0,1], \mathcal{A}}\} \cup \{q_i \mid i \in [d-1]\}, \Sigma_2 \cup \{\star\}, \delta, q_{\emptyset, \mathcal{A}}, \{q_{[0,1], \mathcal{A}}\}),$$

where,  $\delta(q_i, \mathbf{a}) = q_{i+1}$  for each  $\mathbf{a} \in \Sigma_2$ , where  $\delta(q_{[0,1], \mathcal{A}}, \mathbf{a}) = q_{[0,1], \mathcal{A}}$  for each  $\mathbf{a} \in \Sigma_2$  and where  $\delta(q_{\emptyset, \mathcal{A}}, \star) = q_{[0,1], \mathcal{A}}$ . If  $\delta(q, \mathbf{a})$  is not defined above, it is equal to  $q_{\emptyset, \mathcal{A}}$ . This Büchi automaton has  $d + 2$  states and its alphabet has 2 letters, hence its size is  $O(d)$ . The automaton  $\mathcal{A}^{\text{par}}$  is pictured in Figure 3b without its state  $q_{\emptyset, \mathcal{A}}$ . Note that the size of the minimal  $d$ -parallel automaton is exponential in the size of the minimal  $d$ -sequential automaton.

We now explain how to transform a sequential automaton into a parallel one.



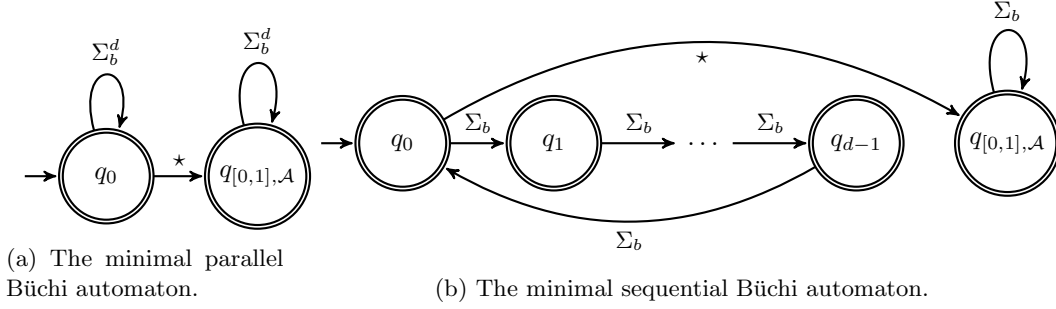


Figure 3: Minimal parallel and sequential RVA accepting  $(\mathbb{R}^{\geq 0})^d$ .

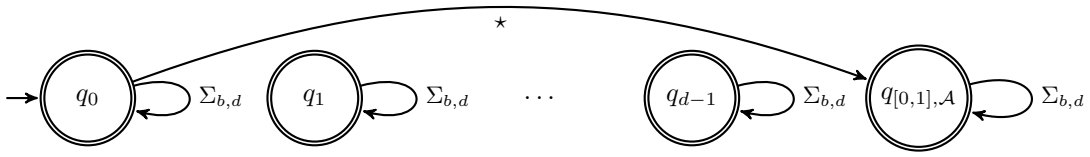


Figure 4: The parallelization of the automaton of Figure 3b.

**Definition 6.3** ( $\text{par}_d(\mathcal{A})$ ). Let  $\mathcal{A} = (Q, \Sigma_b \cup \{\star\}, \delta, q_0, F)$  be a  $d$ -sequential automaton. Let  $\text{par}_d(\delta) : (Q \otimes (\Sigma_{b,d} \cup \{\star\})) \rightarrow Q$  such that  $\text{par}_d(\delta)(q, \star) = \delta(q, \star)$  and such that,  $\text{par}_d(\delta)(q, \mathbf{a}) = \delta(q, a_0 \dots a_{d-1})$  for each  $\mathbf{a} \in \Sigma_{b,d}$ . Then let  $\text{par}_d(\mathcal{A}) = (Q, \Sigma_{b,d} \cup \{\star\}, \text{par}_d(\delta), q_0, F)$ .

This operation is called the *parallelization* of  $\mathcal{A}$ . The parallelization of the automaton pictured in Figure 3b is pictured in Figure 4. We now state two lemmas whose proofs are simple applications of the definitions. Those lemmas show that this notion of parallelization is coherent with the parallelization of words.

**Lemma 6.4.** Let  $w \in \Sigma_b^{d\mathbb{N}} \star \Sigma_b^\omega$  and  $\mathcal{A}$  a  $d$ -sequential automaton. The automaton  $\mathcal{A}$  accepts  $w$  if and only if  $\text{par}_d(\mathcal{A})$  accepts  $\text{par}_d(w)$ .

**Lemma 6.5.** Let  $w \in \Sigma_{b,d}^* \star \Sigma_{b,d}^\omega$  and  $\mathcal{A}$  a  $d$ -sequential automaton. The automaton  $\text{par}_d(\mathcal{A})$  accepts  $w$  if and only if  $\mathcal{A}$  accepts  $\text{seq}_d(w)$ .

Finally, we state that changing the initial state commute with parallelization.

**Lemma 6.6.** Let  $\mathcal{A}$  be a  $d$ -sequential automaton and  $q$  a state of  $\mathcal{A}$ . The automaton  $\text{par}_d(\mathcal{A})_q$  is equal to the automaton  $\text{par}_d(\mathcal{A}_q)$ .

## 6.2 Algorithm

We now consider the problem of deciding whether a Büchi automaton is  $d$ -parallel or  $d$ -sequential. We now state the two main results of this section.

**Theorem 6.7.** Let  $\mathcal{A}$  be an automaton over alphabet  $\Sigma_{b,d} \cup \{\star\}$  with  $n$  states. It is decidable in time  $O(nb^d)$  and space  $O(n)$  whether  $\mathcal{A}$  is a  $d$ -parallel automaton.

**Theorem 6.8.** Let  $\mathcal{A}$  be an automaton over alphabet  $\Sigma_b \cup \{\star\}$  with  $n$  states. It is decidable in time  $O(nbd)$  and space  $O(nd)$  whether  $\mathcal{A}$  is a  $d$ -sequential automaton.

We prove both theorems simultaneously. More precisely, we prove the following proposition:

**Proposition 6.9.** Let  $d_{\text{par}}, d_{\text{seq}} > 0$ . Let  $\mathcal{A}$  be an automaton over alphabet  $\Sigma_{b, d_{\text{par}}} \cup \{\star\}$  with  $n$  states. It is decidable in time  $O(nb^{d_{\text{par}}} d_{\text{seq}})$  and space  $O(nd_{\text{seq}})$  whether  $\mathcal{A}$  recognize a subset of  $\Sigma_{b, d_{\text{par}}}^{d_{\text{seq}}\mathbb{N}} \star \Sigma_{b, d_{\text{par}}}^{d-1}$ .

When  $d_{\text{par}} = d$  and  $d_{\text{seq}} = 1$ , this proposition implies Theorem 6.7, and when  $d_{\text{par}} = 1$  and  $d_{\text{seq}} = d$ , this proposition implies Theorem 6.8. In order to prove this proposition, we introduce the following sets of states.

**Definition 6.10.**  $[Q_\emptyset, Q_F \text{ and } Q_i]$  Let  $\mathcal{A}$  be an automaton over alphabet  $\Sigma_{b, d_{\text{par}}} \cup \{\star\}$ .

- Let  $Q_\emptyset$  be the set of states  $q$  such that  $\mathcal{A}_q$  recognizes the empty language.
- For  $i \in [d_{\text{seq}} - 1]$ , let  $Q_i$  be the set of states  $\delta(q, \mathbf{w})$  with  $\mathbf{w} \in \Sigma_{b, d_{\text{par}}}^{d_{\text{seq}}\mathbb{N}+i}$ .
- Let  $Q_F$  be the set of states  $\delta(q, \mathbf{w})$  with  $\mathbf{w} \in \Sigma_{b, d_{\text{par}}}^* \star \Sigma_{b, d_{\text{par}}}^*$ .

Intuitively, while the automaton reads the natural part of a vector, it visits successively a state of  $Q_0$ , a state of  $Q_1$ , ..., a state of  $Q_{d-1}$ , and then, again a state of  $Q_0$  and so on. Similarly, while the automaton read the fractional part of the word, it visits states of  $Q_F$ . We could prove that, if  $\mathcal{A}$  accepts a subset of  $\Sigma_{b, d_{\text{par}}}^{d_{\text{seq}}\mathbb{N}} \star \Sigma_{b, d_{\text{par}}}^\omega$ , then the intersection of two of those sets is included in  $Q_\emptyset$ . We now give example of those set of states.

**Example 6.11.** Let  $\mathcal{A}$  be the automaton pictured in Figure 3a,  $d_{\text{par}} = d$  and  $d_{\text{seq}} = 1$ . Then  $Q_\emptyset = \{q_{\emptyset, \mathcal{A}}\}$ ,  $Q_0 = \{q_0\}$  and  $Q_F = \{q_{[0,1], \mathcal{A}}\}$ .

Let  $\mathcal{A}$  be the automaton pictured in Figure 3b,  $d_{\text{par}} = 1$  and  $d_{\text{seq}} = 2$ . Then  $Q_\emptyset = \{q_{\emptyset, \mathcal{A}}\}$ ,  $Q_i = \{q_i\}$  and  $Q_F = \{q_{[0,1], \mathcal{A}}, q_{\emptyset, \mathcal{A}}\}$ .

We now characterize the automata accepting a subset of  $\Sigma_{b, d_{\text{par}}}^{d_{\text{seq}}\mathbb{N}} \star \Sigma_{b, d_{\text{par}}}^\omega$  using sets introduced in Definition 6.10. We then characterize those sets of states. All characterizations of those objects are straightforward from their definitions.

**Proposition 6.12.** Let  $\mathcal{A}$  be an automaton over alphabet  $\Sigma_{b, d_{\text{par}}} \cup \{\star\}$ . It accepts a subset of  $\Sigma_{b, d_{\text{par}}}^{d_{\text{seq}}\mathbb{N}} \star \Sigma_{b, d_{\text{par}}}^{d-1}$  if and only if,  $\delta(q, \star) \in Q_\emptyset$  for each state  $q \in Q_F \cup \bigcup_{i=1}^{d-1} Q_i$ .

**Lemma 6.13.** *The set  $Q_\emptyset$  is the greatest set of states included in  $Q$ , which does not contain the accepting recurrent states and which is closed under taking predecessor.*

**Lemma 6.14.** *The sets  $Q_0, \dots, Q_{d_{\text{seq}}-1}$  is the smallest family of sets such that  $q_0 \in Q_0$  and for each  $i \in [d_{\text{seq}} - 1]$ , for each  $q \in Q_i$  and for each  $\mathbf{a} \in \Sigma_{b, d_{\text{par}}}$ , the state  $\delta(q, \mathbf{a})$  belongs to  $Q_{i+1}$ .*

**Lemma 6.15.** *The set  $Q_F$  is the smallest set containing all sets of the form  $\delta(q, \star)$  for  $q \in \bigcup_{i=0}^{d-1} Q_i$  and  $\delta(q, \mathbf{a})$  for  $q \in Q_F$  and  $\mathbf{a} \in \Sigma_{b, d_{\text{par}}}$ .*

It is now explained how to compute efficiently those sets.

**Lemma 6.16.** *All sets of Definition 6.10 are computable in time  $O(nb^{d_{\text{par}}} d_{\text{seq}})$  and space  $O(nd_{\text{seq}})$ .*

*Proof.* Let us first consider the set  $Q_\emptyset$ . The algorithms is a straightforward application of the characterization given in Lemma 6.13. Tarjan's algorithm [Tar72] can be used to compute the set of strongly connected component in time  $O(nb^{d_{\text{par}}})$ , and therefore the set of recurrent states. Furthermore, it is easy to associate in linear time to each state its set of predecessors. Let  $p_q$  be the number of predecessors of a state  $q$ .

Two sets **PotentiallyEmpty** and **ToProcess** are used by the algorithm. The algorithm initializes the set **PotentiallyEmpty** to  $Q$  and initializes the set **ToProcess** to the empty set. The algorithm runs on each recurrent state  $q$ . For each state  $q$ , if  $q$  is accepting, then  $q$  is removed from **PotentiallyEmpty** and added to **ToProcess**. The algorithm then runs on each element  $q$  of **ToProcess**. For each state  $q$ , the algorithm removes  $q$  from **ToProcess** and runs on each predecessors  $q'$  of  $q$ . For each  $q'$ , if  $q'$  is in **PotentiallyEmpty**, then  $q'$  is removed from **PotentiallyEmpty** and added to **ToProcess**. Finally, when **ToProcess** is empty, the algorithm halts and  $Q_\emptyset$  is the value of **PotentiallyEmpty**.

Let us now consider the complexity of this algorithm. At most  $n$  states are added to **ToProcess**, and each state is added at most once. For each state  $q$  added to **ToProcess**, each of its  $p_q$  predecessor is considered in constant time. Thus the algorithm runs in time  $O\left(n + \sum_{q \in Q} p_q\right) = O(nb^{d_{\text{par}}})$ .

The  $d_{\text{seq}}$ -sets  $Q_i$  are computed simultaneously, using the characterization given in Lemma 6.14. The computation of the state  $Q_F$  is similar. The  $d$  sets  $Q_i$  are initialized to the empty set, and **ToProcess** is initialized to  $\{(q_0, 0)\}$ . The algorithm runs on each  $(q, i) \in \text{ToProcess}$ . For each  $(q, i)$ , the pair  $(q, i)$  is removed from **ToProcess** and added into  $Q_i$ . The algorithm runs on each  $\mathbf{a} \in \Sigma_{b, d_{\text{par}}}$ . For each  $\mathbf{a}$ , if  $\delta(q, \mathbf{a}) \notin Q_{(i+1 \bmod d_{\text{seq}})}$ , then  $\delta(q, \mathbf{a})$  is added into **ToProcess**. When **ToProcess** is empty, all states of  $Q_i$  are indeed added to this set.

Let us consider the complexity. Each pair  $(q, i)$  is removed at most once from **ToProcess**, thus the outer loop is executed at most  $O(nd_{\text{seq}})$  times. Each execution of this loop clearly runs in time  $O(b^{d_{\text{par}}})$ . This algorithm stores a number, a state, and  $d_{\text{seq}}$  set of states, thus it clearly takes space  $O(nd_{\text{seq}})$ . □

We now prove Proposition 6.9.

*Proof.* The algorithm computes the sets of Definition 6.10. The algorithm runs on each  $q \in Q_F \cup \bigcup_{i=1}^{d-1} Q_i$  and rejects if  $\delta(q, \star) \notin Q_\emptyset$ . By Proposition 6.12, this algorithm accepts if and only if  $\mathcal{A}$  accepts a subset of  $\Sigma_{b, d_{\text{par}}}^{d_{\text{seq}} \mathbb{N}} \star \Sigma_{b, d_{\text{par}}}^{d-1}$ . By Lemma 6.16, those sets can be computed in time  $O(nb^{d_{\text{par}}} d_{\text{seq}})$  and space  $O(nd_{\text{seq}})$ , and the loops clearly runs in time  $O(n)$  and constant space. Thus, the whole algorithm runs in time  $O(nb^{d_{\text{par}}} d_{\text{seq}})$  and space  $O(nd_{\text{seq}})$ . □

## 7 Fixing a component

In order to consider the two encodings of rational numbers, we must consider the vector of words  $\langle \mathbf{w}, S \rangle$  such that a suffix of some component  $w_f$  belongs to  $0^\omega$  or  $(b-1)^\omega$ . More precisely, at some points of the run, the automaton will only have to  $0^\omega$  or  $(b-1)^\omega$  in some component  $f$ . Since a component is fixed, we may change the alphabet to fix this letter. We do it by replacing this  $f$ -th component by an atomic symbol  $\square$ .

We define a function which remove some component of a word in Section 7.1. We introduce the automata which reads those new words in Section 7.2.

### 7.1 Vector of words

We now introduced a new alphabet. Letters of this alphabet correspond to letters of  $\Sigma_{b, d}$  with some component fixed. We could have considered  $\Sigma_{b, d-1}$ , but this would lead to trouble when  $d = 1$ . Indeed, a word would then be an element of  $()^* \star ()^\omega$ , where  $()$  is the unique 0 tuple, and

it would not be clear what the sequentialization of such a word would be. Instead of removing a component, we choose to replace it with an atomic symbol  $\square$ . The formal definition is now given.

**Definition 7.1** ( $\Sigma_{b,d}^{\square @f}$ ). For  $f \in [d-1]$ , let  $\Sigma_{b,d}^{\square @f} = (\Sigma_{b,f-1} \otimes \{\square\} \otimes \Sigma_{d,b-f})$ .

We now introduce a notation to change a word by fixing one of its component.

**Definition 7.2** ( $\text{fix}^{z @f}(\langle \mathbf{w}, S \rangle)$ ). We first consider  $d$ -parallel words. Let  $f \in [d-1]$ ,  $z \in \Sigma_b \cup \{\square\}$ ,  $\mathbf{w}$  a word whose alphabet is a set of  $d$ -tuples. Let  $S \subseteq \mathbb{N}$ . Let  $\text{fix}^{z @f}(\langle \mathbf{w}, S \rangle)$  be  $\langle \mathbf{w}', S \rangle$ , where  $|\mathbf{w}'| = |\mathbf{w}|$ ,  $w'_f \in z^\infty$ , and  $w'_i = w_i$  for each  $i \neq f$ .

We now consider  $d$ -sequential words. Let  $\langle w, S \rangle \in (\Sigma_b \cup \{\star, \square\})^\infty$ , then  $\text{fix}^{z @f}(\langle w, S \rangle)$  is  $\langle \text{seq}_d(\text{fix}^{z @f}(\text{par}_d(w))), S \rangle$ . Equivalently, this transformation consists in replacing each letter whose position is equivalent to  $f$  modulo  $d$  - not counting the  $\star$ 's - by the letter  $z$ .

The following three lemmas about  $\text{fix}^{z @f}$  are straightforward consequences from its definition.

**Lemma 7.3.** Let  $0 \leq i < f < d$  integers,  $z \in \Sigma_b \cup \{\square\}$ ,  $v \in (\Sigma_b \cup \{\square\})^i$  and  $w \in (\Sigma_b \cup \{\star, \square\})^\infty$ , then  $\text{fix}^{z @f}(vw) = v \text{fix}^{z @f-f-i}(w)$ .

Note that the letter  $\star$  does not belongs to  $v$ . We now state that, if  $w_f \in z^\infty$ , then changing twice position  $f$  is equivalent to doing only the last change.

**Lemma 7.4.** Let  $f \in [d-1]$ ,  $z, z' \in \Sigma_b \cup \{\square\}$ , let  $w$  a  $d$ -parallel or a  $d$ -sequential word. We have  $\text{fix}^{z @f}(\text{fix}^{z' @f}(\langle \mathbf{w}, S \rangle)) = \text{fix}^{z @f}(\langle \mathbf{w}, S \rangle)$ .

We now state that, if  $w_f \in z^\infty$ , then  $\mathbf{w}$  is a fixpoint of the function.

**Lemma 7.5.** Let  $f \in [d-1]$ . Let  $z \in \Sigma_b \cup \{\square\}$ . Let  $\mathbf{w}$  be a  $d$ -parallel word. If  $w_f \in z^\infty$  then  $\text{fix}^{z @f}(\langle \mathbf{w}, S \rangle) = \langle \mathbf{w}, S \rangle$ .

## 7.2 Automata

A notation is now introduced, in order to fix the digit read in some position of an automaton. In this section, we fix  $f \in [d-1]$ ,  $z \in \Sigma_b$ .

**Definition 7.6** ( $\mathcal{A}^{z @f}$ ). Let  $\mathcal{A} = (Q, \Sigma_{b,d} \cup \{\star\}, \delta, q_0, F)$  be a  $d$ -parallel Büchi automaton, then let:

$$\mathcal{A}^{z @f} = (Q, \Sigma_{b,d}^{\square @f} \cup \{\star\}, \delta^{z @f}, q_0, F),$$

where  $\delta^{z @f}(q, \mathbf{a}) = \delta(q, \text{fix}^{z @f}(\mathbf{a}))$  for all  $\mathbf{a} \in \Sigma_{b,d}^{\square @f} \cup \{\star\}$ .

**Definition 7.7** ( $\mathcal{A}^z$ ). Let  $\mathcal{A} = (Q, \Sigma_b \cup \{\star\}, \delta, q_0, F)$  be a  $d$ -sequential Büchi automaton. Let:

$$\mathcal{A}^z = (Q', \Sigma_b \cup \{\star, \square\}, \delta^z, (q_0, 0), F),$$

where  $Q' = (Q \otimes [d-1]) \cup \{q_{\emptyset, \mathcal{A}}\}$  and where, for each state  $q \in Q$ ,  $\delta^z((q, i), \star) = (\delta(q, \star), i)$ ,  $\delta^z((q, i), a) = (\delta(q, a), i+1)$  for each  $a \in \Sigma_b$  and  $i \in \{0, \dots, d-2\}$ ,  $\delta^z((q, d-1), \square) = (\delta(q, z), 0)$ . For each state  $q \in Q'$  and each letter  $a \in \Sigma_b \cup \{\star, \square\}$  such that  $\delta^z(q, a)$  is not yet defined, it is set to  $q_{\emptyset, \mathcal{A}}$ .

It is now stated that the two transformations introduced above preserve weakness.

**Lemma 7.8.** *Let  $\mathcal{A}$  be a weak  $d$ -parallel automaton, then the automaton  $\mathcal{A}^{z@f}$  is weak. Let  $\mathcal{A}$  be a weak  $d$ -sequential automaton, then the automaton  $\mathcal{A}^z$  is weak.*

*Proof.* For the case of  $d$ -parallel automata, it suffices to remark that each strongly connected component of  $\mathcal{A}^{z@f}$  is a subset of a strongly connected component of  $\mathcal{A}$ . For the case of  $d$ -sequential automata, it suffices to remark that a strongly connected component of  $\mathcal{A}^z$  is either  $\{q_{\emptyset, \mathcal{A}}\}$ , or a set  $S$  such that  $\{q \mid (q, i) \in S\}$  is a strongly connected component of  $\mathcal{A}$ .  $\square$

We also state that changing the initial state of a  $d$ -parallel automaton commute with the transformation introduced above.

**Lemma 7.9.** *Let  $\mathcal{A}$  be a  $d$ -parallel automaton and  $q$  a state of  $\mathcal{A}$ . Then  $(\mathcal{A}^{z@f})_q = (\mathcal{A}_q)^{z@f}$ .*

**Lemma 7.10.** *Let  $\mathcal{A}$  be a  $d$ -sequential automaton and  $q$  a state of  $\mathcal{A}$ . Then  $(\mathcal{A}^z)_{(q,0)} = (\mathcal{A}_q)^z$ .*

**Words and automata** We show, in this section, how the notations introduced in the two preceding sections interact. The first two lemmas deal with replacing a component with a  $\square$ .

**Lemma 7.11.** *Let  $\mathcal{A}$  be a  $d$ -parallel automaton. Let  $\langle \mathbf{w}, S \rangle \in (\Sigma_{b,d} \cup \{\star\})^\omega$  such that  $w_f = z^\omega$ . The automaton  $\mathcal{A}$  accepts  $\langle \mathbf{w}, S \rangle$  if and only if  $\mathcal{A}^{z@f}$  accepts  $\text{fix}^{\square@f}(\langle \mathbf{w}, S \rangle)$ .*

**Lemma 7.12.** *Let  $\mathcal{A}$  be a  $d$ -sequential automaton. Let  $\langle w, S \rangle \in \Sigma_b \cup \{\star\}^\omega$  such that  $\text{par}_d(w)_{d-1} = z^\omega$ . The automaton  $\mathcal{A}$  accepts  $\langle w, S \rangle$  if and only if  $\mathcal{A}^z$  accepts  $\text{fix}^{\square@d-1}(\langle w, S \rangle)$ .*

The following lemma deals with replacing a component of a vector by a single letter.

**Lemma 7.13.** *Let  $\mathcal{A}$  be a  $d$ -parallel automaton. Let  $\langle \mathbf{w}, S \rangle \in (\Sigma_{b,d}^{\square@f} \cup \{\star\})^\omega$ . The automaton  $\mathcal{A}^{z@f}$  accepts  $\langle \mathbf{w}, S \rangle$  if and only if  $\mathcal{A}$  accepts  $\text{fix}^{z@f}(\langle \mathbf{w}, S \rangle)$ .*

The following lemma illustrates how the notation introduced above behaves on a word with a component whose suffix is  $z^\omega$ .

**Lemma 7.14.** *Let  $\mathcal{A} = (Q, \Sigma_b \cup \{\star\}, \delta, q_0, F)$ ,  $q \in Q$ ,  $\langle \mathbf{v}, V \rangle \in (\Sigma_{b,d} \cup \{\star\})^*$  and  $\langle \mathbf{w}, W \rangle \in (\Sigma_{b,d} \cup \{\star\})^\omega$  with  $w_f = z^\omega$ . Then  $\mathcal{A}$  accepts  $\langle \mathbf{v}, V \rangle \langle \mathbf{w}, W \rangle$  if and only if  $(\mathcal{A}^{z@f})_{\delta(q, \langle \mathbf{v}, V \rangle)}$  accepts  $\text{fix}^{\square@f}(\langle \mathbf{w}, W \rangle)$ .*

Note that, in the last term, the function  $\delta$  is still the transition function of  $\mathcal{A}$  and not the one of  $\mathcal{A}^{z@f}$ .

*Proof.* The fact “ $\mathcal{A}$  accepts  $\langle \mathbf{v}, V \rangle \langle \mathbf{w}, W \rangle$ ” is equivalent to “ $\mathcal{A}_{\delta(q_0, \langle \mathbf{v}, V \rangle)}$  accepts  $\langle \mathbf{w}, W \rangle$ ”. By Lemma 7.11, since  $w_f = z^\omega$ , those statements are equivalent to “ $(\mathcal{A}_{\delta(q_0, \langle \mathbf{v}, V \rangle)})^{z@f}$  accepts  $\text{fix}^{\square@f}(\langle \mathbf{w}, W \rangle)$ ”. By Lemma 7.8, they are also equivalent to: “ $(\mathcal{A}^{z@f})_{\delta(q, \langle \mathbf{v}, V \rangle)}$  accepts  $\text{fix}^{\square@f}(\langle \mathbf{w}, W \rangle)$ ”.  $\square$

## 8 Characterizations of $d$ -parallel RVA

Recall from the introduction that a RVA is a Büchi automaton accepting a saturated language. In this section, we give some characterizations of the  $d$ -parallel RVAs. The last of those characterization allows us to give in Section 10 an algorithm which decides whether an automaton over alphabet  $\Sigma_{b,d} \cup \{\star\}$  is a RVA. We use the other characterizations to prove the last one. We first prove a property of minimal RVA.

**Lemma 8.1.** Let  $\mathcal{A} = (Q, A, \delta, q_0, F)$  a minimal  $d$ -parallel RVA, then  $\delta(q_0, \mathbf{0}) = q_0$ .

*Proof.* Since  $\mathcal{A}$  is minimal, it suffices to prove  $L_\omega(\mathcal{A}_{\delta(q_0, \mathbf{0})}) = L_\omega(\mathcal{A})$ , hence it suffices to prove that  $\mathcal{A}$  accepts  $\mathbf{w}$  if and only if  $\mathcal{A}_{\delta(q_0, \mathbf{0})}$  for all  $\mathbf{w} \in (\Sigma_{b,d} \cup \{\star\})^\omega$ .

Let  $\mathbf{w} \in (\Sigma_{b,d} \cup \{\star\})^\omega$ . The automaton  $\mathcal{A}_{\delta(q_0, \mathbf{0})}$  accepts  $\mathbf{w}$  if and only if  $\mathcal{A}$  accepts  $\mathbf{0w}$ . Since  $\mathcal{A}$  is saturated and  $[\mathbf{0w}]_b^\mathbb{R} = [\mathbf{w}]_b^\mathbb{R}$ ,  $\mathcal{A}$  accepts  $\mathbf{0w}$  if and only if it accepts  $\mathbf{w}$ . By transitivity of equivalence,  $\mathcal{A}$  accepts  $\mathbf{w}$  if and only if  $\mathcal{A}_{\delta(q_0, \mathbf{0})}$  accepts  $\mathbf{w}$ .  $\square$

We now state the characterizations.

**Proposition 8.2.** Let  $\mathcal{A}$  be a minimal weak  $d$ -parallel Büchi automaton such that  $\delta(q_0, \mathbf{0}) = q_0$ . The following statement are equivalent:

1. The automaton  $\mathcal{A}$  is a RVA.
2. For all  $\langle \mathbf{w}, \{s\} \rangle \langle \mathbf{w}', \{s\} \rangle \in \Sigma_{b,d}^* \star \Sigma_{b,d}^\omega$  such that  $[\langle \mathbf{w}, \{s\} \rangle]_b^\mathbb{R} = [\langle \mathbf{w}', \{s\} \rangle]_b^\mathbb{R}$  and such that  $\mathcal{A}$  accepts  $\langle \mathbf{w}, \{s\} \rangle$ ,  $\mathcal{A}$  accepts  $\langle \mathbf{w}', \{s\} \rangle$ .
3. for all  $\langle \mathbf{w}, \{s\} \rangle \langle \mathbf{w}', \{s\} \rangle \in \Sigma_{b,d}^* \star \Sigma_{b,d}^\omega$  such that  $[\langle \mathbf{w}, \{s\} \rangle]_b^\mathbb{R} = [\langle \mathbf{w}', \{s\} \rangle]_b^\mathbb{R}$ , such that  $|\{j \mid w_j \neq w'_j\}| = 1$ , and such that  $\mathcal{A}$  accepts  $\langle \mathbf{w}, \{s\} \rangle$ ,  $\mathcal{A}$  accepts  $\langle \mathbf{w}', \{s\} \rangle$ .
4. for each  $f \in [d-1]$ , for each  $q \in Q$ , accessible in  $\mathcal{A}$  from  $q_0$ , for each  $\mathbf{a} \in \Sigma_{b,d}$  with  $a_f < b-1$ ,

$$L_\omega\left(\left(\mathcal{A}^{(b-1)@f}\right)_{\delta(q, \mathbf{a})}\right) = L_\omega\left(\left(\mathcal{A}^{0@f}\right)_{\delta(q, \mathbf{a}')}\right), \quad (3)$$

where  $\mathbf{a}' = a_0 \dots a_{f-1}(a_f + 1)a_{f+1} \dots a_{d-1}$ .

Note that Property (1) requires to consider  $d$ -tuple of words, with  $\star$ 's in potentially different positions, while in Property (2), both words have a single  $\star$ 's at the same position. Property (2) requires to consider any pair of word whose natural part have the same length, while Property (3) restrict our study to the case where all but one components of the word are equal.

*Proof.* The proof is done by the following sequence of implications. Property (1) implies Property (4), which implies Property (3), which implies Property (2), which implies Property (1).

**Property (1) implies Property (4)** Let  $f, q, \mathbf{a}, \mathbf{a}'$  as in the hypothesis. Let us prove that  $L_\omega\left(\left(\mathcal{A}^{(b-1)@f}\right)_{\delta(q, \mathbf{a})}\right) \subseteq L_\omega\left(\left(\mathcal{A}^{0@f}\right)_{\delta(q, \mathbf{a}')}\right)$ , the proof of the reverse inclusion is similar. Let  $\langle \mathbf{w}, T \rangle \in \left(\Sigma_{b,d}^{\square @f} \cup \{\star\}\right)^\omega$  accepted by  $\left(\mathcal{A}^{(b-1)@f}\right)_{\delta(q, \mathbf{a})}$ , let us prove that it is accepted by  $\left(\mathcal{A}^{0@f}\right)_{\delta(q, \mathbf{a}')}$ .

Since  $\left(\mathcal{A}^{(b-1)@f}\right)_{\delta(q, \mathbf{a})}$  accepts  $\langle \mathbf{w}, T \rangle$ , by Lemma 7.13,  $\mathcal{A}_{\delta(q, \mathbf{a})}$  accepts  $\text{fix}^{b-1@f}(\langle \mathbf{w}, T \rangle)$ . Since  $q$  is accessible from  $q_0$ , there exists  $\langle \mathbf{v}, S \rangle \in (\Sigma_{b,d} \cup \{\star\})^*$  such that  $\delta(q_0, \langle \mathbf{v}, S \rangle) = q$ . Since  $\mathcal{A}_{\delta(q, \mathbf{a})}$  accepts  $\text{fix}^{b-1@f}(\langle \mathbf{w}, T \rangle)$  and  $\delta(q_0, \langle \mathbf{v}, S \rangle) = q$ , it follows that  $\mathcal{A}$  accepts  $\langle \mathbf{v}, S \rangle \mathbf{a} \text{fix}^{b-1@f}(\langle \mathbf{w}, T \rangle)$ . Let us prove that  $\left[\langle \mathbf{v}, S \rangle \mathbf{a} \text{fix}^{b-1@f}(\langle \mathbf{w}, T \rangle)\right]_b^\mathbb{R} = \left[\langle \mathbf{v}, S \rangle \mathbf{a}' \text{fix}^{0@f}(\langle \mathbf{w}, T \rangle)\right]_b^\mathbb{R}$ . That is, for  $j \in [d-1]$ , we want to prove that  $\left[\left(\langle \mathbf{v}, S \rangle \mathbf{a} \text{fix}^{b-1@f}(\langle \mathbf{w}, T \rangle)\right)\right]_i^\mathbb{R} = \left[\left(\langle \mathbf{v}, S \rangle \mathbf{a}' \text{fix}^{0@f}(\langle \mathbf{w}, T \rangle)\right)\right]_i^\mathbb{R}$ . For  $i \in [d-1] \setminus f$ , it suffices to remark that:

$$\left(\langle \mathbf{v}, S \rangle \mathbf{a} \text{fix}^{b-1@f}(\langle \mathbf{w}, T \rangle)\right)_i = \langle v_i, S \rangle a_i \langle w_i, T \rangle = \langle v_i, S \rangle a'_i \langle w_i, T \rangle = \left(\langle \mathbf{v}, S \rangle \mathbf{a}' \text{fix}^{0@f}(\langle \mathbf{w}, T \rangle)\right)_i.$$

It remains to consider the case  $i = f$ . Remark that

$$\left( \langle \mathbf{v}, S \rangle \mathbf{a} \text{ fix}^{b-1 \oplus f}(\langle \mathbf{w}, T \rangle) \right)_f = \langle v_f, S \rangle a_f \langle (b-1)^\omega, T \rangle$$

and

$$\left( \langle \mathbf{v}, S \rangle \mathbf{a}' \text{ fix}^{0 \oplus f}(\langle \mathbf{w}, T \rangle) \right)_f = \langle v_f, S \rangle a'_f \langle 0^\omega, T \rangle = \langle v_f, S \rangle (a_f + 1) \langle 0^\omega, T \rangle.$$

By Theorem 3.1, both of those words encode the same number.

Since  $\mathcal{A}$  accepts  $\langle \mathbf{v}, S \rangle \mathbf{a} \text{ fix}^{b-1 \oplus f}(\langle \mathbf{w}, T \rangle)$ ,  $\left[ \langle \mathbf{v}, S \rangle \mathbf{a} \text{ fix}^{b-1 \oplus f}(\langle \mathbf{w}, T \rangle) \right]_b^{\mathbb{R}} = \left[ \langle \mathbf{v}, S \rangle \mathbf{a}' \text{ fix}^{0 \oplus f}(\langle \mathbf{w}, T \rangle) \right]_b^{\mathbb{R}}$  and  $\mathcal{A}$  is saturated,  $\mathcal{A}$  accepts  $\langle \mathbf{v}, S \rangle \mathbf{a}' \text{ fix}^{0 \oplus f}(\langle \mathbf{w}, T \rangle)$ . It follows that  $\mathcal{A}_{\delta(q, \mathbf{a}')} \text{ accepts } \text{fix}^{0 \oplus f}(\langle \mathbf{w}, T \rangle)$ . Finally, since  $\mathcal{A}_{\delta(q, \mathbf{a}')} \text{ accepts } \text{fix}^{0 \oplus f}(\langle \mathbf{w}, T \rangle)$ , by Lemma 7.13,  $(\mathcal{A}^{0 \oplus f})_{\delta(q, \mathbf{a}')} \text{ accepts } \langle \mathbf{w}, T \rangle$ .

**Property (4) implies Property (3)** Let  $\langle \mathbf{w}, \{s\} \rangle$  and  $\langle \mathbf{w}', \{s\} \rangle$  be two words as in Property (3). Let us prove that  $\mathcal{A}$  accepts  $\mathbf{w}'$ . Let  $f \in [d-1]$  be the only integer such that  $w_f \neq w'_f$ . As explained in Theorem 3.1, since  $[\langle w_f, \{s\} \rangle]_b^F = [\langle w'_f, \{s\} \rangle]_b^F$ , since  $w_f \neq w'_f$ , and since their natural parts have the same length, then  $\{w_f, w'_f\} = \{u_f a_f (b-1)^\omega, u_f (a_f + 1) 0^\omega\}$ , for some  $u_f \in \Sigma_b^*$  and  $a_f \in \Sigma_b \setminus \{b-1\}$ . Let us assume that  $w_f = u_f a_f (b-1)^\omega$ , the case where  $w_f = u_f (a_f + 1) 0^\omega$  is similar. Note that  $w'_f = u_f (a_f + 1) 0^\omega$ .

Let  $l$  be the length of the prefix of  $(w_f, \{s\})$  before the occurrence of the letter  $a_f$ . It is  $|u_f|$  if  $s > |u_f|$ , and  $|u_f| + 1$  otherwise. Let  $\langle \mathbf{u}, U \rangle = \langle \mathbf{w}, \{s\} \rangle[< l] = \langle \mathbf{w}', \{s\} \rangle[< l]$  be this prefix. Let  $\mathbf{a} = \langle \mathbf{w}, \{s\} \rangle[l]$  and  $\mathbf{a}' = \langle \mathbf{w}', \{s\} \rangle[l]$ . Similarly, let  $\langle \mathbf{v}, V \rangle = \langle \mathbf{w}, \{s\} \rangle[\geq l+1]$  and  $\langle \mathbf{v}', V \rangle = \langle \mathbf{w}', \{s\} \rangle[\geq l+1]$  be the suffixes of  $\mathbf{w}$  and  $\mathbf{w}'$  after those occurrences of  $\mathbf{a}$  and of  $\mathbf{a}'$  respectively. Note that the notations  $u_f$ ,  $a_f$  and  $a'_f$  introduced above are consistent with the notations  $\mathbf{u}$ ,  $\mathbf{a}$  and  $\mathbf{a}'$ . Note also that  $v_f = (b-1)^\omega$ , that  $v'_f = 0^\omega$ , and that  $v_i = v'_i$  for all  $i \in [d-1] \setminus \{f\}$ . Hence  $\text{fix}^{\square \oplus f}(\langle \mathbf{v}, V \rangle) = \text{fix}^{\square \oplus f}(\langle \mathbf{v}', V \rangle)$ . The notation  $a_f$  introduced above is coherent with the notation  $\mathbf{a}$ . Since  $a'_f = a_f + 1$ ,  $\mathbf{a}$  and  $\mathbf{a}'$  satisfy the hypothesis of Property (4). It follows that  $L_\omega((\mathcal{A}^{(b-1) \oplus f})_{\delta(q, \mathbf{a})}) = L_\omega((\mathcal{A}^{0 \oplus f})_{\delta(q, \mathbf{a}')}).$

We can now prove that  $\langle \mathbf{w}', \{s\} \rangle$  is accepted by  $\mathcal{A}$ . Since  $\langle \mathbf{w}, \{s\} \rangle = \langle \mathbf{u}, U \rangle \mathbf{a} \langle \mathbf{v}, V \rangle$  is accepted by  $\mathcal{A}$  and  $v_f = (b-1)^\omega$ , by Lemma 7.14,  $(\mathcal{A}^{(b-1) \oplus f})_{\delta(q_0, \langle \mathbf{u}, U \rangle \mathbf{a})} \text{ accepts } \text{fix}^{\square \oplus f}(\langle \mathbf{v}, V \rangle)$ . Since  $(\mathcal{A}^{(b-1) \oplus f})_{\delta(q_0, \langle \mathbf{u}, U \rangle \mathbf{a})} \text{ accepts } \text{fix}^{\square \oplus f}(\langle \mathbf{v}, V \rangle)$  and  $L_\omega((\mathcal{A}^{(b-1) \oplus f})_{\delta(q, \mathbf{a})}) = L_\omega((\mathcal{A}^{0 \oplus f})_{\delta(q, \mathbf{a}')}).$ ,  $(\mathcal{A}^{0 \oplus f})_{\delta(q_0, \langle \mathbf{u}, U \rangle \mathbf{a}')} \text{ accepts } \text{fix}^{\square \oplus f}(\langle \mathbf{v}, V \rangle)$ . Since  $\text{fix}^{\square \oplus f}(\langle \mathbf{v}, V \rangle) = \text{fix}^{\square \oplus f}(\langle \mathbf{v}', V \rangle)$  and  $(\mathcal{A}^{0 \oplus f})_{\delta(q_0, \langle \mathbf{u}, U \rangle \mathbf{a}')} \text{ accepts } \text{fix}^{\square \oplus f}(\langle \mathbf{v}, V \rangle)$ ,  $(\mathcal{A}^{0 \oplus f})_{\delta(q_0, \langle \mathbf{u}, U \rangle \mathbf{a}')} \text{ accepts } \text{fix}^{\square \oplus f}(\langle \mathbf{v}', V \rangle)$ . It follows from Lemma 7.14 that  $\mathcal{A}$  accepts  $\langle \mathbf{u}, U \rangle \mathbf{a}' \langle \mathbf{v}, V \rangle = \langle \mathbf{w}', \{s\} \rangle$ .

**Property (3) implies Property (2)** We must prove that, for all  $\langle \mathbf{w}, \{s\} \rangle, \langle \mathbf{w}', \{s\} \rangle \in (\Sigma_{b,d} \cup \{\star\})^\omega$ , if  $\mathcal{A}$  accepts  $\langle \mathbf{w}, \{s\} \rangle$  and  $[\langle \mathbf{w}, \{s\} \rangle]_b^{\mathbb{R}} = [(\mathbf{w}', \{s\})]_b^{\mathbb{R}}$  then  $\mathcal{A}$  accepts  $\langle \mathbf{w}', \{s\} \rangle$ . The proof is by induction on  $i = |\{j \mid w_j \neq w'_j\}|$ . The case  $i = 0$  is trivial, since it means that  $\langle \mathbf{w}, \{s\} \rangle = \langle \mathbf{w}', \{s\} \rangle$ . Let us now assume that  $i > 1$  and that the induction hypothesis holds when  $|\{j \mid w_j \neq w'_j\}| < i$ .

Since  $|\{j \mid w_j \neq w'_j\}| = i$  and  $i > 1$ , there exists  $f \in [d-1]$  such that  $w_f \neq w'_f$ . Let  $\langle \mathbf{w}'', \{s\} \rangle \in (\Sigma_{b,d} \cup \{\star\})^\omega$  such that  $w''_f = w_f$  and such that  $w''_k = w'_k$  for all  $k \in [b-1] \setminus \{f\}$ . Note that  $|\{j \mid w_j \neq w''_j\}| = i-1$ ,  $|\{j \mid w'_j \neq w''_j\}| = 1$  and  $[\langle \mathbf{w}, \{s\} \rangle]_b^{\mathbb{R}} = [\langle \mathbf{w}'', \{s\} \rangle]_b^{\mathbb{R}} = [\langle \mathbf{w}', \{s\} \rangle]_b^{\mathbb{R}}$ .

Since  $|\{j \mid w_j \neq w''_j\}| = i-1$  and  $[\langle \mathbf{w}, \{s\} \rangle]_b^{\mathbb{R}} = [\langle \mathbf{w}'', \{s\} \rangle]_b^{\mathbb{R}}$ , by induction hypothesis  $\mathcal{A}$  accepts  $\langle \mathbf{w}'', \{s\} \rangle$ . Since  $|\{j \mid w''_j \neq w'_j\}| = 1$  and  $[\langle \mathbf{w}'', \{s\} \rangle]_b^{\mathbb{R}} = [\langle \mathbf{w}', \{s\} \rangle]_b^{\mathbb{R}}$  and since  $\mathcal{A}$  accepts  $\langle \mathbf{w}'', \{s\} \rangle$ , by Property (3),  $\mathcal{A}$  accepts  $\langle \mathbf{w}', \{s\} \rangle$ .

**Property (2) implies Property (1)** We must prove that, for all  $\langle \mathbf{w}, \{s\} \rangle, (\mathbf{w}', \{s'\}) \in (\Sigma_{b,d} \cup \{\star\})^\omega$ , if  $\mathcal{A}$  accepts  $\langle \mathbf{w}, \{s\} \rangle$  and  $[\langle \mathbf{w}, \{s\} \rangle]_b^\mathbb{R} = [\langle \mathbf{w}', \{s'\} \rangle]_b^\mathbb{R}$  then  $\mathcal{A}$  accepts  $\langle \mathbf{w}', \{s'\} \rangle$ . Let us assume that  $s \leq s'$ , the case  $s > s'$  is similar. Since  $\delta(q_0, \mathbf{0}) = q_0$  and  $\mathcal{A}$  accepts  $\langle \mathbf{w}, \{s\} \rangle$ ,  $\mathcal{A}$  accepts  $\langle \mathbf{0}^{s'-s}, \{s'\} \rangle$ . Note that  $[\langle \mathbf{0}^{s'-s}, \{s'\} \rangle]_b^\mathbb{R} = [\langle \mathbf{w}, \{s\} \rangle]_b^\mathbb{R} = [\langle \mathbf{w}', \{s'\} \rangle]_b^\mathbb{R}$ . Since  $[\langle \mathbf{0}^{s'-s}, \{s'\} \rangle]_b^\mathbb{R} = [\langle \mathbf{w}', \{s'\} \rangle]_b^\mathbb{R}$  and  $\mathcal{A}$  accepts  $\langle \mathbf{0}^{s'-s}, \{s'\} \rangle$ , by Property (2),  $\mathcal{A}$  accepts  $\langle \mathbf{w}', \{s'\} \rangle$ .  $\square$

## 9 Characterization of $d$ -sequential automata

A characterization of  $d$ -sequential automata is now given. This characterization is similar to Property 4 of Proposition 8.2. Instead of doing the whole proof again for sequential automata, we prove that this characterization is correct by proving that it implies the characterization of Proposition 8.2 on the parallelization of the  $d$ -sequential automata.

**Proposition 9.1.** Let  $\mathcal{A}$  be a minimal weak  $d$ -sequential Büchi automaton. The following statements are equivalent:

1. The automaton  $\mathcal{A}$  is a RVA.
2.
  - $\delta(q_0, 0^d) = q_0$  and
  - For each  $q \in Q$  accessible in  $\mathcal{A}$  from  $q_0$ , for each  $a \in \Sigma_b \setminus b - 1$ ,  $(\mathcal{A}^{(b-1)})_{(\delta(q,a),0)}$  and  $(\mathcal{A}^0)_{(\delta(q,a+1),0)}$  accept the same language.

In order to prove this proposition, we introduce the following lemma. This lemma allows us to reduce Property (2) of Proposition 9.1 to Property (4) of Proposition 8.2.

**Lemma 9.2.** Let  $\mathcal{A}$  be a weak  $d$ -sequential Büchi automaton,  $f \in [d-1]$ ,  $z \in \Sigma_b$ ,  $q$  a state of  $\mathcal{A}$ ,  $\mathbf{a} \in \Sigma_{b,d}$  and  $\langle \mathbf{w}, S \rangle \in (\Sigma_{b,d}^{\square @ f} \cup \{\star\})^\omega$ . The word  $\langle \mathbf{w}, S \rangle$  is accepted by  $(\text{par}_d(\mathcal{A})^{z @ f})_{(\text{par}_d(\delta)(q, \mathbf{a}), 0)}$  if and only if  $(\mathcal{A}^z)_{(\delta(q, a_0 \dots a_f))}$  accepts  $a_{f+1} \dots a_{d-1} \text{seq}_d(\langle \mathbf{w}, S \rangle)$ .

*Proof.* By Lemma 7.10 “ $(\text{par}_d(\mathcal{A})^{z @ f})_{(\text{par}_d(\delta)(q, \mathbf{a}), 0)}$  accepts  $\langle \mathbf{w}, S \rangle$ ” is equivalent to: “ $(\text{par}_d(\mathcal{A})_{\text{par}_d(\delta)(q, \mathbf{a})})^{z @ f}$  accepts  $\langle \mathbf{w}, S \rangle$ ”. By Lemma 7.13, it is equivalent to: “ $(\text{par}_d(\mathcal{A})_{\text{par}_d(\delta)(q, \mathbf{a})})$  accepts  $\text{fix}^{z @ f}(\langle \mathbf{w}, S \rangle)$ ”, hence to “ $(\text{par}_d(\mathcal{A}))_q$  accepts  $\mathbf{a} \text{fix}^{z @ f}(\langle \mathbf{w}, S \rangle)$ ”. By Lemma 6.6, it is also equivalent to: “ $\text{par}_d(\mathcal{A}_q)$  accepts  $\mathbf{a} \text{fix}^{z @ f}(\langle \mathbf{w}, S \rangle)$ ”. By Lemma 6.5, it is also equivalent to: “ $\mathcal{A}_q$  accepts  $\text{seq}_d(\mathbf{a} \text{fix}^{z @ f}(\langle \mathbf{w}, S \rangle))$ ”. Note that  $\text{seq}_d(\mathbf{a} \text{fix}^{z @ f}(\langle \mathbf{w}, S \rangle))$  equals  $a_0 \dots a_{d-1} \text{seq}_d(\text{fix}^{z @ f}(\langle \mathbf{w}, S \rangle))$ , thus the above-mentioned properties are equivalent to: “ $\mathcal{A}_q$  accepts  $a_0 \dots a_{d-1} \text{seq}_d(\text{fix}^{z @ f}(\langle \mathbf{w}, S \rangle))$ ” and then to: “ $\mathcal{A}_{\delta(q, a_0 \dots a_f)}$  accepts  $a_{f+1} \dots a_{d-1} \text{seq}_d(\text{fix}^{z @ f}(\langle \mathbf{w}, S \rangle))$ ”. By Lemma 7.12, it is equivalent to: “ $(\mathcal{A}_{\delta(q, a_0 \dots a_f)})^z$  accepts  $\text{fix}^{\square @ d-1}(a_{f+1} \dots a_{d-1} \text{seq}_d(\text{fix}^{z @ f}(\langle \mathbf{w}, S \rangle)))$ ”.

Note that the length of  $a_{f+1} \dots a_{d-1}$  is  $(d-1) - (f+1) + 1 = d - f - 1$ , and that  $(d-1) - (d-f-1) = f$ , thus by Lemma 7.3,  $\text{fix}^{\square @ d-1}(a_{f+1} \dots a_{d-1} \text{seq}_d(\text{fix}^{z @ f}(\langle \mathbf{w}, S \rangle)))$  equals  $a_{f+1} \dots a_{d-1} \text{fix}^{\square @ f}(\text{seq}_d(\text{fix}^{z @ f}(\langle \mathbf{w}, S \rangle)))$ . By definition of  $\text{fix}^{\square @ f}$  on  $d$ -sequential number,



$\text{fix}^{\square @ f}(\text{seq}_d(\text{fix}^{z @ f}(\langle \mathbf{w}, S \rangle)))$  equals  $\text{seq}_d(\text{fix}^{\square @ f}(\text{fix}^{z @ f}(\langle \mathbf{w}, S \rangle)))$ . By Lemma 7.4,  $\text{fix}^{\square @ f}(\text{fix}^{z @ f}(\langle \mathbf{w}, S \rangle))$  equals  $\text{fix}^{\square @ f}(\langle \mathbf{w}, S \rangle)$ . Since  $w_f = \square^\omega$ , by Lemma 7.5,  $\text{fix}^{\square @ f}(\langle \mathbf{w}, S \rangle) = \langle \mathbf{w}, S \rangle$ . It follows that  $\text{fix}^{\square @ d-1}(a_{f+1} \dots a_{d-1} \text{seq}_d(\text{fix}^{z @ f}(\langle \mathbf{w}, S \rangle)))$  equals  $a_{f+1} \dots a_{d-1} \text{seq}_d(\langle \mathbf{w}, S \rangle)$ . Thus, all the above mentioned facts are equivalent to: “ $(\mathcal{A}_{\delta(q, a_0 \dots a_f)})^{b-1}$  accepts  $a_{f+1} \dots a_{d-1} \text{seq}_d(\langle \mathbf{w}, S \rangle)$ ”. Finally, by Lemma 7.9, it is equivalent to “ $(\mathcal{A}^z)_{\delta(q, a_0 \dots a_f)}$  accepts  $a_{f+1} \dots a_{d-1} \text{seq}_d(\langle \mathbf{w}, S \rangle)$ ”.  $\square$

Proposition 9.1 is now proven.

*Proof.* Let us show that Property (1) implies Property (2). The proof of the first part of this Property is the same than the proof of Lemma 8.1. The proof of the second part of this Property is the same than the proof that Property (1) of Proposition 8.2 implies Property (4) of Proposition 8.2.

It remains to prove that Property (2) implies Property (1). We want to prove that  $\mathcal{A}$  is a RVA. Since  $\mathcal{A}$  is  $d$ -sequential, it remains to prove that  $L_\omega(\mathcal{A})$  is saturated. It suffices to prove that  $\text{par}_d(L_\omega(\mathcal{A}))$  is saturated. By Lemma 6.4,  $\text{par}_d(\mathcal{A})$  accepts  $\text{par}_d(L_\omega(\mathcal{A}))$ . Therefore, we only have to prove that  $\text{par}_d(\mathcal{A})$  is a RVA. By Proposition 8.2, it suffices to prove that  $\text{par}_d(\delta)(q_0, \mathbf{0}) = q_0$  and that, for each  $f \in [d-1]$ , for each  $q \in Q$ , accessible in  $\mathcal{A}$  from  $q_0$ , for each  $\mathbf{a} \in \Sigma_{b,d}$  with  $a_f < b-1$ ,  $L_\omega\left(\left(\text{par}_d(\mathcal{A})^{(b-1) @ f}\right)_{\text{par}_d(\delta)(q, \mathbf{a})}\right) = L_\omega\left(\left(\text{par}_d(\mathcal{A})^{0 @ f}\right)_{\text{par}_d(\delta)(q, \mathbf{a}')} \right)$  where  $\mathbf{a}' = a_0 \dots a_{f-1}(a_f + 1)a_{f+1} \dots a_{d-1}$ .

Note that the first part of Property (2) clearly implies  $\text{par}_d(\delta)(q_0, \mathbf{0}) = q_0$ . Let  $q, \mathbf{a}, \mathbf{a}'$  as above, it remains to prove that  $L_\omega\left(\left(\text{par}_d(\mathcal{A})^{(b-1) @ f}\right)_{\text{par}_d(\delta)(q, \mathbf{a})}\right) \subseteq L_\omega\left(\left(\text{par}_d(\mathcal{A})^{0 @ f}\right)_{\text{par}_d(\delta)(q, \mathbf{a}')} \right)$ , the reverse inclusion is similar. Let  $\langle \mathbf{w}, S \rangle \in \Sigma_{b,d}^{\square @ f} \cup \{\star\}^\omega$  accepted by  $\left(\text{par}_d(\mathcal{A})^{(b-1) @ f}\right)_{\text{par}_d(\delta)(q, \mathbf{a})}$ , we want to prove that it is accepted by  $\left(\text{par}_d(\mathcal{A})^{0 @ f}\right)_{\text{par}_d(\delta)(q, \mathbf{a}')}.$

Since  $\left(\text{par}_d(\mathcal{A})^{(b-1) @ f}\right)_{\text{par}_d(\delta)(q, \mathbf{a})}$  accepts  $\langle \mathbf{w}, S \rangle \in \left(\Sigma_{b,d}^{\square @ f} \cup \{\star\}\right)^\omega$ , by Lemma 9.2,  $(\mathcal{A}_{\delta(q, a_0 \dots a_{f-1} a_f)})^{b-1}$  accepts  $a_{f+1} \dots a_{d-1} \text{seq}_d(\langle \mathbf{w}, S \rangle)$ . By the second part of Property (2), it follows that  $(\mathcal{A}_{\delta(q, a_0 \dots a_{f-1} (a_f + 1))})^0$  accepts  $a_{f+1} \dots a_{d-1} \text{seq}_d(\langle \mathbf{w}, S \rangle)$ . By Lemma 9.2, it follows that  $\left(\text{par}_d(\mathcal{A})^{0 @ f}\right)_{\text{par}_d(\delta)(q, \mathbf{a}')} \text{ accepts } \langle \mathbf{w}, S \rangle$ .  $\square$

## 10 Algorithms

We now show how to decide efficiently whether an automaton is a RVA.

**Theorem 10.1.** *Let  $\mathcal{A} = (Q, \Sigma_{b,d} \cup \{\star\}, \delta, q_0, F)$  an automaton with  $n$  states. It is decidable in time  $O(n \log(n) db^d)$  and space  $O(nb^d)$  whether  $\mathcal{A}$  is a RVA.*

Note that  $b^d$  is the cardinality of the alphabet. Thus this algorithm is quasi-linear in the size of its input.

*Proof.* Without loss of generality, it can be assumed that the automaton is minimal. The algorithm consists in three parts. First, the algorithm checks whether the algorithm of Theorem 6.7 applied on  $\mathcal{A}$  returns true. Secondly, the algorithm checks whether  $\delta(q_0, \mathbf{0}) = q_0$ . Thirdly, the algorithm runs on each  $f \in [d-1]$ . For each  $f$ , the algorithm generates the automata  $\mathcal{A}^{0 @ f}$  and  $\mathcal{A}^{(b-1) @ f}$  and the data structure mentioned in Corollary 4.5. The algorithm runs on each

$q \in Q$  and  $\mathbf{a}, \mathbf{a}'$  as in Property (4) of Proposition 8.2. The algorithm then applies the algorithm of Corollary 4.5 to check whether, for all pairs  $(\text{par}_d(\delta)(q, \mathbf{a}), \text{par}_d(\delta)(q, \mathbf{a}'))$ ,  $(\mathcal{A}^{b-1 \otimes f})_{\text{par}_d(\delta)(q, \mathbf{a})}$  and  $(\mathcal{A}^{0 \otimes f})_{\text{par}_d(\delta)(q, \mathbf{a}')}$  accept the same language. If one of those checks fail the algorithm rejects. Otherwise, the algorithm accepts.

By Lemma 7.8,  $\mathcal{A}^{0 \otimes f}$  and  $\mathcal{A}^{(b-1) \otimes f}$  are weak, thus, Corollary 4.5 can be used. It follows from Theorem 6.7, Lemma 8.1 and Proposition 8.2 that this algorithm accepts exactly the  $d$ -parallel automata which are RVA.

It remains to consider the complexity. By Theorem 6.7, the first part runs in  $O(nb^d)$  and space  $O(n)$ . The second part clearly runs in constant time and space. By Corollary 4.5, the last part runs in time  $O(n \log(n)b^d d)$  and space  $O(n \log nb^d)$ .  $\square$

We now prove a similar theorem for  $d$ -sequential automata.

**Theorem 10.2.** *Let  $\mathcal{A} = (Q, \Sigma_b \cup \{\star\}, \delta, q_0, F)$  an automaton with  $n$  states. It is decidable in time  $O(nd \log(nd)b)$  and space  $O(ndb)$  whether  $\mathcal{A}$  is a RVA.*

*Proof.* Without loss of generality, it can be assumed that the automaton is minimal. The algorithm and its proof are similar to the ones of Theorem 10.1. The algorithm consists in three parts. Firstly, the algorithm applies the algorithm of Theorem 6.8 to check whether  $\mathcal{A}$  is a  $d$ -sequential automaton. Secondly, it checks whether  $\delta(q_0, 0^d) = q_0$ . Thirdly, the algorithm generates the automata  $\mathcal{A}^0$  and  $\mathcal{A}^{(b-1)}$  and the data structure mentionned in Corollary 4.5. The algorithm runs on each  $q \in Q$  and  $a \in \Sigma_b \setminus b - 1$ . For each  $q$  and  $a$ , the algorithm applies the algorithm of Corollary 4.5 to check whether  $(\mathcal{A}^{(b-1)})_{\delta(q, a)}$  and  $(\mathcal{A}^0)_{\delta(q, a+1)}$  accept the same language.

By Lemma 7.8,  $\mathcal{A}^0$  and  $\mathcal{A}^{(b-1)}$  are weak, thus, Corollary 4.5 can be used. It follows from Theorem 6.8 and Proposition 9.1 that this algorithm accepts exactly the  $d$ -sequential automata which are RVAs.

It remains to consider the complexity. By Theorem 6.8, the first part runs in time  $O(ndb)$  and space  $O(nd)$ . The second part runs in time  $O(d)$  and constant space. Finally, by Corollary 4.5, the third part runs in time  $O(nd \log(nd)b)$  and space  $O(ndb)$ .  $\square$

A last algorithm is given for the special case of dimension  $d = 1$ .

**Theorem 10.3.** *Let  $\mathcal{A} = (Q, \Sigma_b, \delta, q_0, F)$  a minimal weak Büchi automaton with  $n$  states. It is decidable in time  $O(nb)$  and space  $O(n)$  whether  $L_\omega(\mathcal{A})$  is saturated.*

Note that  $\mathcal{A}$  is assumed minimal.

*Proof.* Note that  $\Sigma_{b,1}^{\square \otimes f}$  contains exactly one letter,  $(\square)$ , thus both languages of Equation (3) are either  $\emptyset$  or  $(\square)^\omega$ . It follows that Equation (3) holds if and only if  $\delta(q, a) \in Q_\emptyset^{(b-1)}$  is equivalent to  $\delta(q, a+1) \in Q_\emptyset^0$ , where  $Q_\emptyset^z$  is the set of states  $q$  such that  $\mathcal{A}_q^{z \otimes 0}$  accepts the empty language.

The algorithm is now given. The algorithm checks whether  $\delta(q_0, 0) = q_0$ . The algorithm computes  $\mathcal{A}^{(b-1) \otimes 0}$  and  $\mathcal{A}^{0 \otimes 0}$ . The algorithm applies the algorithm of Lemma 6.16 to those two automata in order to compute the sets  $Q_\emptyset^{(b-1)}$  and  $Q_\emptyset^0$ . The algorithm runs on each state  $q \in Q$  accessible from  $q_0$ , and on each  $a \in \Sigma_b \setminus \{b-1\}$ . For each  $q$  and  $a$ , the algorithm checks whether  $\delta(q, a) \in Q_\emptyset^{(b-1)}$  is equivalent to  $\delta(q, a+1) \in Q_\emptyset^0$ . Finally, if one of the checks fail, the algorithm rejects, otherwise it accepts.

Let us consider the complexity. The automata  $\mathcal{A}^{(b-1) \otimes 0}$  and  $\mathcal{A}^{0 \otimes 0}$  clearly takes space  $O(n)$ . Applying the algorithm of Lemma 6.16 takes time  $O(n)$  and takes space  $O(n)$ . For a set  $q$  and a letter  $a$ , checking the equivalence is done in constant time and space. Thus, the final loop runs

in time  $O(nb)$  and constant space. Finally, the whole algorithm runs in time  $O(nb)$  and space  $O(n)$ .  $\square$

## 11 Considering negative reals

In this section, we consider the case of negative numbers. Given  $w_I \in \Sigma_b^*$ ,  $aw_I$  encodes, in  $b$ -complement representation, the number  $[w_I]_b^I$  if  $a = 0$ ,  $-b^{|w_I|} + [w_I]_b^I$  if  $a = (b-1)$ , and is undefined otherwise. Similarly, given  $w_F \in \Sigma_b^\omega$ ,  $[aw_I \star w_F]_b^{\mathbb{R}}$  encodes, in  $b$ -complement representation, the number encoded by  $aw_I$ , plus  $[w_F]_b^F$ .

When considering  $b$ -complement representation, Theorem 3.1 must be changed as follows. For  $l$  great enough, a real  $q$  has two encoding whose natural part's length is  $l$  if and only if it is of the form  $nb^p$  with  $n, p \in \mathbb{Z}$ . If  $q = 0$ , those encodings are  $(b-1)^l \star (b-1)^\omega$  and  $0^l \star 0^\omega$ , otherwise they are as in Equation (1).

A characterization of automata accepting saturated languages in  $b$ -complement representation is now given. This characterization and its proof is similar to the ones of (8.2).

**Proposition 11.1.** Let  $\mathcal{A} = (Q, \Sigma_{b,d} \cup \{\star\}, \delta, q_0, F)$  a weak Büchi automaton over alphabet  $\Sigma_{b,d} \cup \{\star\}$ . It accepts a saturated language in  $b$ -complement if and only if:

1.  $\delta(q_0, \mathbf{a}\mathbf{a}) = \delta(q_0, \mathbf{a})$  for all  $\mathbf{a} \in \{0, (b-1)\}^d$ ,
2.  $\delta(q_0, \mathbf{a}) \in Q_\emptyset$  for all  $\mathbf{a} \in \Sigma_{b,d} \cup \{\star\} \setminus \left(\{0, (b-1)\}^d\right)$ ,
3. for each  $f \in [d-1]$ , for each  $q \in Q \setminus q_0$ , accessible in  $\mathcal{A}$  from  $q_0$ , for each  $\mathbf{a} \in \Sigma_{b,d}$  with  $a_f < b-1$ ,

$$L_\omega\left(\left(\mathcal{A}^{(b-1)@f}\right)_{\delta(q, \mathbf{a})}\right) = L_\omega\left(\left(\mathcal{A}^{0@f}\right)_{\delta(q, \mathbf{a}')} \right),$$

where  $\mathbf{a}' = a_0 \dots a_{f-1}(a_f + 1)a_{f+1} \dots a_{d-1}$

- 4.

$$L_\omega\left(\left(\mathcal{A}^{(b-1)@f}\right)\right) = L_\omega\left(\left(\mathcal{A}^{0@f}\right)\right),$$

Property (4) allows to consider the case of the real 0. Note that Property (1) considers  $b^d$  letters. It implies that this proposition does not lead to a polynomial time algorithm in the case of  $d$ -sequential automata. In the case of  $d$ -parallel automata, this proposition leads easily to algorithms, as Proposition 9.1 led to Theorem 10.1 and to Theorem 10.3.

**Theorem 11.2.** Let  $\mathcal{A} = (Q, \Sigma_{b,d} \cup \{\star\}, \delta, q_0, F)$  an automaton with  $n$  states reading reals in  $b$ -complement. It is decidable in time  $O(n \log(n)db^d)$  and space  $O(nb^d)$  whether  $\mathcal{A}$  is a RVA.

**Theorem 11.3.** Let  $\mathcal{A} = (Q, \Sigma_b \cup \{\star\}, \delta, q_0, F)$  a minimal weak Büchi automaton with  $n$  states reading reals in  $b$ -complement. It is decidable in time  $O(nb)$  and space  $O(n)$  whether  $L_\omega(\mathcal{A})$  accepts a saturated language in  $b$ -complement.

## 12 Conclusion

In this paper, we have proven that it is decidable in quasi-linear time whether a weak Büchi automaton reading digits and dots accept a language which encode a saturated set of vector reals.

Two natural questions remain open.

Can this algorithm be adapted for some classes of automata which are not weak. Even in the case of dimension 1, it seems complicated to test whether  $L_\omega(\mathcal{A}_{\delta(q,0)}) = L_\omega(\mathcal{A})$ , when the automaton is not weak.

Given an automaton  $\mathcal{A}$  which accept a set  $R \subseteq (\mathbb{R}^{\geq 0})^d$ , is there some efficient way to compute a saturated automaton  $\mathcal{A}'$  which also accept  $R$ . One could compute a  $\text{FO}[\mathbb{R}, \mathbb{Z}; X_b, +, <]$ -formula defining  $R$ , and from this formula a saturated Büchi automaton. However, this method is inefficient, and does not preserve weakness.

## References

- [BBB10] Bernard Boigelot, Julien Brusten, and Véronique Bruyère. On the sets of real numbers recognized by finite automata in multiple bases. *Logical Methods in Computer Science*, 6(1), 2010.
- [BBL09] Bernard Boigelot, Julien Brusten, and Jérôme Leroux. A Generalization of Semenov’s Theorem to Automata over Real Numbers. In Renate A. Schmidt, editor, *Automated Deduction, 22nd International Conference, CADE 2009*, volume 5663 of *Lecture Notes in Computer Science*, pages 469–484, Montréal, Canada, August 2009. Springer Berlin.
- [BHMV94] Véronique Bruyère, Georges Hansel, Christian Michaux, and Roger Villemaire. Logic and p-recognizable sets of integers. *Bull. Belg. Math. Soc.*, 1:191–238, 1994.
- [Cob69] Alan Cobham. On the base-dependence of sets of numbers recognizable by finite automata. *Mathematical systems theory*, 3(2):186–192, 1969.
- [HW60] Godfrey Harold Hardy and Edward Maitland Wright. An introduction to the theory of numbers, 1960. Autres tirages avec corrections : 1962, 1965, 1968, 1971, 1975.
- [Lö01] Christof Löding. Efficient minimization of deterministic weak omega automata, 2001.
- [Sem77] A. L. Semenov. The presburger nature of predicates that are regular in two number systems. *Siberian Math. J.*, 18:289,299, 1977.
- [Tar72] Robert Tarjan. Depth-first search and linear graph algorithms. *SIAM Journal on Computing*, 1(2):146–160, 1972.
- [WB00] Pierre Wolper and Bernard Boigelot. On the construction of automata from linear arithmetic constraints. In Susanne Graf and Michael Schwartzbach, editors, *Tools and Algorithms for the Construction and Analysis of Systems*, volume 1785 of *Lecture Notes in Computer Science*, pages 1–19. Springer Berlin Heidelberg, 2000.

- |   |   |
|---|---|
| <ul style="list-style-type: none"> <li>• <math>\omega</math>-word, 3</li> <li>• <math>d</math>-parallel language, 5</li> <li>• <math>d</math>-sequential language, 5</li> <li>• <math>L^*</math>, 3</li> <li>• <math>L^{d\mathbb{N}+j}</math>, 3</li> </ul> | <ul style="list-style-type: none"> <li>• <math>L^+</math>, 3</li> <li>• <math>[\cdot]_b^F</math>, 4</li> <li>• <math>[\cdot]_b^I</math>, 4</li> <li>• <math>[\cdot]_b^{\mathbb{R}}</math>, 4</li> <li>• <math>\mathbf{0}</math>, 3</li> </ul> |
|---|---|

- $\mathcal{A}_q$  for  $\mathcal{A}$  an automaton and  $q$  a state., 6
- $L_\omega(\mathcal{A})$ , 6
- $[\mathcal{A}]_b^{\mathbb{R}}$ , 8
- Accessible, 6
- $\mathcal{A}^z$  for  $\mathcal{A}$  a  $d$ -sequential Büchi automaton, 12
- $\mathcal{A}^{z@f}$  for  $\mathcal{A}$  a  $d$ -parallel Büchi automaton, 12
- Alphabet, 3
- Concatenation of words, 3
- Deterministic Büchi automaton, 6
- $\epsilon$ , 3
- Finite word, 3
- Fractional part of a word of  $\Sigma_b^* \star \Sigma_b^\omega$ , 4
- $[L]_b^{\mathbb{R}}$  for  $L$  a  $d$ -parallel language., 5
- Language, 3
- $\omega$ -language, 3
- Letter, 3
- Morphism of automata, 6, 7
- $\mathbb{N}$ , 3
- $[n]$  for  $n \in \mathbb{N}$ ., 3
- Natural part of a word of  $\Sigma_b^* \star \Sigma_b^\omega$ , 4
- $\text{par}_d(\langle w, S \rangle)$ , 5
- Parallelization of a word, 5
- Parallelization of an automaton, 9
- $\text{par}_d(L)$  for  $L$  a  $d$ -sequential language, 5
- Predecessor state, 6
- $w[< n]$  - Prefix of  $w$  of length  $n$ , 3
- $Q_F$ , 10
- $Q_\emptyset$ , 10
- Quotient of automata, 6
- $\mathbb{R}$ , 3
- Run, 6
- Run of an automaton, 6
- Saturated language, 5
- $\text{seq}_d(\mathbf{w})$ , 5
- Sequentialization of a word, 5
- $\Sigma_{b,d}^{\square@f}$ , 12
- Strongly connected components, 6
- $w[\geq n]$  - Suffix of  $w$  without its  $n$  first letters., 3
- $\text{fix}^{z@f}(\langle \mathbf{w}, S \rangle)$ , 12